2

RSSP- A FORTRAN SIMULATION PACKAGE FOR USE IN TEACHING RESPONSE SURFACE METHODOLOGY

James T. Treharne

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama

June 12, 1991

RSSP- A FORTRAN SIMULATION PACKAGE FOR USE IN TEACHING RESPONSE SURFACE METHODOLOGY

James T. Treharne

Certificate of Approval:

Bruce E. Herring, Co-Chair
Professor
Industrial Engineering

Saeed Maghsoodloo

Saeed Maghsoodloo, Co-Chair
Professor
Industrial Engineering

James N. Hool

James N. Hool
Professor
Industrial Engineering

Norman J. Doorenbos

Norman J. Doorenbos
Dean
Graduate School

Style manual or journal used IE TRANSACTIONS

Computer software used Word Perfect 5.1

RSSP- A FORTRAN SIMULATION PACKAGE
FOR USE IN TEACHING RESPONSE
SURFACE METHODOLOGY

James T. Treharne

A Thesis
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the
Degree of
Master of Science

Auburn, Alabama
June 12, 1991

RSSP- A FORTRAN SIMULATION PACKAGE
FOR USE IN TEACHING RESPONSE
SURFACE METHODOLOGY

James T. Treharne

James T Treharne
Signature of Author

April 9, 1991
Date

Copy sent to:

Name	Date

THESIS ABSTRACT

RSSP- A FORTRAN SIMULATION PACKAGE FOR USE IN TEACHING RESPONSE SURFACE METHODOLOGY

James T. Treharne

Master of Science, June 12, 1991
(B.S., USMA, 1979)

105 Typed Pages

Directed by Saeed Maghsoodloo
and Bruce E. Herring

The Response Surface Simulation Package (RSSP) consists of three Fortran programs that assist in the teaching of Response Surface Methodology. The programs operate on an IBM (or compatible) personal computer. The package helps bridge the gap between theory and practice which is often difficult to do in a classroom setting. The thesis details the background and objectives of the computer package and a review of Response Surface Methodology theory. The simulation package assumes the user has a sufficient background in experimental design, multiple linear regression, and analysis of variance. The thesis also includes the Fortran source code for the programs as well as detailed instructor and student manuals. Further, sample outputs from the three programs are provided. A major objective of this work is to make the programs user friendly, for both the instructor and student. This allows the student to gain a great deal of knowledge about practical experimental design and Response Surface Methodology.

TABLE OF CONTENTS

Appendices

LIST OF TABLES

I. BACKGROUND AND OBJECTIVES

Auburn University's Industrial Engineering department has offered for many years a series of three progressive graduate courses in the design and analysis of experiments. In the third course, IE 632, Response Surface Methodology (RSM) is a primary topic of study. In the early 1970's it was recognized that there was a strong need to give the students the opportunity to apply their knowledge of RSM to a relatively simple, yet realistic problem. Jesse L. Martin, an Auburn graduate student at that time, developed a computer program package called RSAP- Response Surface Analysis Program [3]. RSAP enabled the student to learn a great deal about the practical application of RSM without having to spend an excessive amount of time doing lengthy and repetitive computations. RSAP has proven to be a highly successful tool of reinforcing response surface methodology through a comprehensive practical exercise. The RSAP simulation program continues in use today at Auburn University.

RSAP allows the instructor to load the equations of at most 15 unique surfaces into the computer. The student is then required, without knowledge of the true surface equation, to use proper RSM techniques to arrive near the optimum region, where the assigned surface is estimated by a second-order equation. The student also uses two supplementary programs to find the optimal point (either a maximum or a minimum) on the surface and to draw a series of response contours. The students analyze a surface with two independent variables although there is a capability to analyze as many as five independent variables. The students spend approximately one-half of the quarter investigating their surface. The students must have a thorough knowledge of statistical techniques taught in previous courses in order to complete their investigations.

With the advent of microcomputers in the 1980's, the RSAP program has become, in some respects, outdated. RSAP was designed for use on an IBM mainframe computer during the "punch card" days. Therefore, although RSAP retains its usefulness for instruction, it has become difficult to use by today's standards. The program continues to be run on a mainframe computer. Therefore, the department must expend resources to keep the program loaded on the computer as well as for processing time when the students conduct their investigations. An instructor is not likely to add or change the surface equations loaded on the computer due to the time and effort required to do so. Additionally, the student must go to a mainframe computer terminal to execute his experiments. Each time he executes an experiment, the student must pick up the results at a print station. Therefore, the student may spend a considerable amount of time working on things that are not directly related to RSM. Further, RSAP has no interactive capability and the instructor and student manuals were written nearly twenty years ago. Therefore, the system has grown to be more and more user unfriendly compared to what has become the norm with today's personal computers.

The major objective of this thesis is to develop a completely new version of RSAP which incorporates all the previous benefits while eliminating the current weaknesses. The Response Surface Simulation Package (RSSP) is a complete rework of Mr. Martin's original efforts. RSSP is also written in the Fortran programming language and uses RSAP as a framework. The new package is designed to be both interactive and user friendly. The instructor can easily add, change, or delete surfaces to be studied. The instructor can continue to specify the size of the normal random errors that the simulator uses when calculating the response at various design points. The student can study the surface on any IBM compatible computer. The student can also print experimental results with equal ease. Additionally, the user manuals make the package very easy to use. The complete simulation package enables the student to complete the entire response surface methodology process with maximum learning benefit.

II. REVIEW OF RESPONSE SURFACE METHODOLOGY

The primary goal of Response Surface Methodology is to find the set of conditions which optimize a given response surface. The number of independent variables which may affect a response (dependent) variable range from one upward. The optimal response may be either a maximum or a minimum value. In most real world engineering applications, the experimenter has little idea about the exact relationship between the response variable and the independent variables. The experimenter may not even know which independent variables have a statistically significant impact on the response. Because of the infinite amount of possible arrangements between various variables, RSM was developed to find the optimal set of conditions as quickly and as cost efficiently as possible. RSM is an iterative technique that takes the experimenter from an arbitrary starting point to a local optimum. After finding the optimal set of conditions, the experimenter will have a much greater confidence about the expected output as well as the range of conditions which produce a desired level of output.

Formally, the experimenter desires to find the values for $X_1, X_2, \ldots, X_p$ that maximize (or minimize) a response variable, Y.

$$Y=f(X_1,X_2,\ldots,X_p)$$

During experimentation, the response values at given points will vary because of experimental error. These errors are assumed normally distributed with a mean of zero. As mentioned, RSM is an iterative technique and may require many repetitions before the optimal conditions are found. The general steps are:

1. Design a 1st-order experiment. The design must include a sufficient number of observation points to estimate the regression coefficients, the experimental error, and to test for goodness of fit. It should also use the proper spacing, which depends heavily on experimental error.
2. Conduct first-order experiments.
3. Determine if a first-order model is adequate. If an adequate fit exists and at least one independent variable is significant, move along the path of steepest ascent (descent) to the next center point. If there is not a good fit, adjust the spacing until a satisfactory fit is obtained. If a good fit is impossible and the coefficients remain insignificant, it is time to try a second-order model. On the other hand, if there is a

good fit but no significant coefficients, the experimenter should increase the spacing. It is time to move to a second-order model when the experimenter can achieve a good fit but cannot achieve significant coefficients.

4. Design second-order experiments. There must be a sufficient number of observations as in the first-order design.
5. Conduct the second-order experiments.
6. Determine if there is an adequate fit. If there is a good fit and the coefficients are significant, expand the spacing until the F(LOF) is 75% of the critical F value. If there is not a good fit, the experimenter must decrease the spacing. If the fit is adequate and there are no significant coefficients, the spacing must be increased.
7. Estimate the optimal value for the dependent variable and the values of the independent variables where the optimal occurs.
8. Map contours of the response variable.

Several points must be taken into consideration during the optimization process. First, the experimental design must be properly constructed in order to ensure that the formulas used to derive the analysis of variance results are correct. In the case of first-order experimentation, the designs should be balanced. In the case of second-order models, the designs should be rotatable. The bibliography contains an excellent reference to study Response Surface Methodology and the design of experiments [4]. An additional source highlights an excellent strategy for progressing through the eight steps mentioned above [2]. Second, the correct spacing is critical during experimentation. Experience will help one select an initial spacing. If the spacing is too wide, it will be difficult to get an adequate least-squares fit. On the other hand, if the spacing is too narrow (relative to σ_ϵ), the resulting estimates of the coefficients may not be precise. However, the coefficients may appear to be insignificant because the experimental error is large relative to the size of the spacing. Third, the experimenter may have to conduct many iterations of first-order experiments until the general area of the optimum is reached. The experimenter must be careful about selecting the size of moves that is made along the path of steepest ascent (descent). If the moves are too small, one will expend more money and time to reach the optimum. If the moves are too large, one may inadvertently bypass the optimal point. In summary, second-order experimentation is warranted when the first-order model no longer provides a good fit after appropriate spacing adjustment. At this point, the experimenter should be in the vicinity of the optimum. The second-order model will require a different design than the first order. This is true because three levels of each variable must be examined in order to check for quadratic effects. At this point, the primary concern is to fit a second-order model over the largest possible area. This allows the experimenter to plot the response contours (if desired) over a large area.

III. RSM SOLUTION METHODS

Response Surface Methodology is a procedure used to find the mathematical relationship between a dependent variable, Y, and a number of independent variables- $X_1, X_2, \ldots, X_p$. The methodology provides a sequence to quickly establish that relationship. The mathematical techniques used in the methodology are quite common. Standard linear regression techniques are used to determine the relationship between the variables. A standard analysis of variance (ANOVA) table is then constructed to determine the statistical significance of the coefficients as well as the adequacy of the model being used.

CALCULATION OF REGRESSION COEFFICIENTS

The first step is to estimate the various regression coefficients in the first-order model. The model, in general, is:

$$Y=\beta_o+ \beta_1X_1+ \beta_2X_2+\ldots+\beta_pX_p+ \epsilon \ ,$$

where ϵ represents the experimental error. This error is assumed to be normally distributed with a mean of zero. A simple regression problem with two independent variables is presented to illustrate the computation of the coefficients. If there are "n" total observations, then the following two matrices are written:

$$\mathbf{Y}= \begin{bmatrix} Y_1 \\ Y_2 \\ . \\ . \\ Y_n \end{bmatrix} \qquad \mathbf{X}= \begin{bmatrix} 1 & X_{11} & X_{21} \\ 1 & X_{12} & X_{22} \\ . & . & . \\ . & . & . \\ 1 & X_{1n} & X_{2n} \end{bmatrix}$$

The left hand column of **X** consists of the value of 1 only. These are dummy variables which are associated with β_o. The method of least squares is used to determine the coefficients. Therefore, the goal is to minimize the least squares function:

$$L= \sum_{j=1}^{n} (\epsilon_j)^2 = \sum_{j=1}^{n} (y_j-\beta_o-\beta_1 X_1-\beta_2 X_2)^2$$

In matrix notation the least squares function becomes:

$$L= \epsilon^t * \epsilon= (\mathbf{Y}-\mathbf{X}\beta)^t * (\mathbf{Y}-\mathbf{X}\beta) \quad ,$$

where t stands for transpose. The next step is to take the partial derivatives of the least squares function with respect to each of the coefficients $(\beta_o, \beta_1, \beta_2)$. All three of the partial derivatives are then set equal to zero to obtain the normal equations. In matrix notation, these equations can be rearranged in the form of

$$(\mathbf{X}^t * \mathbf{X}) * \beta = \mathbf{X}^t * \mathbf{Y}.$$

This is equivalent to:

$$\begin{bmatrix} n & \Sigma X_1 & \Sigma X_2 \\ \Sigma X_1 & \Sigma(X_1)^2 & \Sigma X_1 X_2 \\ \Sigma X_2 & \Sigma X_1 X_2 & \Sigma(X_2)^2 \end{bmatrix} * \begin{bmatrix} \beta_o \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \Sigma y \\ \Sigma X_1 y \\ \Sigma X_2 y \end{bmatrix}$$

If $\mathbf{A}= \mathbf{X}^t*\mathbf{X}$ and $\mathbf{G}= \mathbf{X}^t*\mathbf{Y}$, then the solution vector becomes:

$$\beta= \mathbf{A}^{-1}*\mathbf{G}$$

ANALYSIS OF VARIANCE

After the coefficients have been found, the next step is to conduct the analysis of variance. The purpose of this is twofold. First, it is necessary to determine if the coefficients are significant. Second, it is necessary to determine if the first (or second)-order model is satisfactory. The sum of the squares terms used in the analysis of variance are calculated in the following manner.

1. Uncorrected SS(Total)= $\Sigma(y_i^2)$. This is the value of all "n" observations squared.
2. $SS(\beta_o)= (\Sigma y_i)^2/n$. This is commonly referred to as the correction factor.
3. $SS(\beta_1)= (\beta_1)^2/(A_{22})^{-1}$. This is the sum of squares due to β_1 after assuming that the other variables are in the model. In other words, it is the net contribution to the regression sum of the squares.
4. $SS(\beta_2)= (\beta_2)^2/(A_{33})^{-1}$. Same as in previous step.

5. SS(Residual)= SS(Total)-SS(β_o)-SS(β_1)-SS(β_2). The residual sum of the squares accounts for experimental error and error due to the inadequacy of the model.
6. SS(Pure Error)= $\Sigma_k(\Sigma_i y_{ik}^2 - (\Sigma_i y_{ik})^2/h_k)$; where i=1,2,...,h. There are "k" distinct design points and "h_k" observations at the kth design point. This sum of the squares is due to the experimental error when conducting multiple repetitions at a point. In order to conduct F-tests, SS(Error) should have at least five degrees of freedom.
7. SS(Lack of Fit)= SS(Residual) - SS(Pure Error) This error accounts for the inadequacy of the model.

After the above SS's are calculated, the ANOVA table is constructed and the appropriate F-tests made. The procedures for finding the regression coefficients and analysis of variance table for a second-order model closely parallel the method for the first-order model. The second-order model is:

$$Y= \beta_o+ \beta_1X_1+ \beta_2X_2+ \beta_{11}X_1^2+ \beta_{22}X_2^2+ \beta_{12}X_1X_2+ \epsilon$$

Although the experimental design must provide sufficient df to evaluate the quadratic effects, the solution procedure is the same. The analysis of variance table for a first-order model with two independent variables is shown in Table 1.

SOURCE	DF	SS	MS	F-RATIO
TOTAL	n	SS(T)	SS(T)/n	-----
β_o	1	$SS(\beta_o)$	$SS(\beta_o)$	$MS(\beta_o)/MSE$
β_1	1	$SS(\beta_1)$	$SS(\beta_1)$	$MS(\beta_1)/MSE$
β_2	1	$SS(\beta_2)$	$SS(\beta_2)$	$MS(\beta_2)/MSE$
RESIDUAL	n-3	SS(RES)	SS(RES)/ DF(RES)	MS(RES)/MSE
LACK OF FIT	(n-3)- $\Sigma(h_k-1)$	SS(LOF)	SS(LOF)/ DF(LOF)	MS(LOF)/MSE
EXPERI- MENTAL ERROR	$\Sigma(h_K-1)$	SS(EE)	SS(EE)/ DF(EE)	-- ---

Table 1. Analysis of Variance with Two Ind. Variables

DETERMINATION OF CRITICAL VALUES

After a second-order model has been found that best fits the surface, the experimenter must determine the optimal response and the point at which it occurs. The procedure is quite simple. Again, the second-order model is:

$$Y= \beta_o+ \beta_1X_1+ \beta_2X_2+ \beta_{11}X_1{}^2+ \beta_{22}X_2{}^2+ \beta_{12}X_1X_2+ \epsilon$$

The optimum occurs at the point where the partial derivative with respect to each independent variable is equal to zero. In this case:

$$\frac{\partial Y}{\partial X_1} = \beta_1 + 2\beta_{11}X_1 + \beta_{12}X_2 = 0$$

$$\frac{\partial y}{\partial X_2} = \beta_2 + 2\beta_{22}X_2 + \beta_{12}X_1 = 0$$

In matrix form:

$$\begin{bmatrix} 2\beta_{11} & \beta_{12} \\ \beta_{12} & 2\beta_{22} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} -\beta_1 \\ -\beta_2 \end{bmatrix}$$

Clearly, the optimal point is obtained from:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 2\beta_{11} & \beta_{12} \\ \beta_{12} & 2\beta_{22} \end{bmatrix}^{-1} * \begin{bmatrix} -\beta_1 \\ -\beta_2 \end{bmatrix}.$$

The values of X_1 and X_2 are substituted back into the model. Then the optimal value of the response variable is estimated. This is normally a maximum or minimum value. However, it can also be a saddle point. Therefore, in order to determine the exact nature of the optima, one must conduct a canonical analysis [4].

IV. RSSP PROGRAMMING

The Response Surface Simulation Package consists of three executable Fortran programs. The first program, RSMSU.EXE, is used by the instructor to set up a data file for student use. Prior to running this program, the instructor must have previously constructed a data file entitled INSTR.DAT. The second program, RSM.EXE, is the primary program in the simulation. This program produces the simulated response values for each experiment as well as the resulting ANOVA table. The final program, CRIT.EXE, is used to calculate the optimal conditions of the second-order equation, to estimate the optimal value of the dependent variable, and to provide data to map response contours. These programs use RSAP as a general framework. In some segments there is a close parallel to RSAP while in other segments there is no resemblance at all.

This chapter discusses the programming logic in each of the three executable Fortran programs. First, one must understand the capabilities of the simulation package.

1. The simulation package is designed to accommodate at most five independent variables. It is recommended that most introductory courses in RSM,

at least initially, use only two independent variables.

2. The instructor may provide data for at most fifteen different response surfaces. This normally enables each student to experiment with a unique surface.
3. The experimental error is assumed normally distributed with a mean equal to zero. The instructor must specify the size of the error variance.
4. The maximum number of observations is restricted to sixty. That is to say, the degrees of freedom for the corrected total SS's cannot exceed 59. This number is sufficient for almost any practical situation because an experimenter should always design an experiment that minimizes cost while providing the necessary degrees of freedom.
5. Each surface that the instructor inputs is represented by a second-order equation. Some coefficients may be set equal to zero. Its general form is

$$Y = \beta_o + \sum_{i=1}^{5} \beta_{ii} X_i^2 + \sum_{i=1}^{4} \sum_{j=i+1}^{5} \beta_{ij} X_i X_j + \sum_{i=1}^{5} \beta_i X_i .$$

6. The student is responsible for preparing a proper experimental design. Failure to do so may provide

invalid results. For example, if the student uses an unbalanced design, the equations used to calculate the various sum of the squares terms are invalid.

RSMSU.EXE

This program reads an instructor prepared data file (INSTR.DAT). This file contains all the vital data for at most fifteen response surfaces. The program then uses this data file to create an encoded data file (STU.DAT). The student uses STU.DAT when he executes the main program. This program is used to ensure that the student does not have access to the uncoded surface data in the INSTR.DAT. Prior to executing this setup program, the instructor must use a text editor to create an ASCII file called INSTR.DAT. Each surface contains the following six lines of information.

1. Coefficients of the second-order terms, β_{ii}.
2. Coefficients of the interaction terms, β_{ij}.
3. Coefficients of the first-order terms, β_i.
4. Constant term, β_o.
5. Experimental error variance.
6. Center point for the first set of experiments.

RSMSU.EXE first reads and stores the data from INSTR.DAT. The program then encodes the data by adding a constant to each coefficient and then multiplying by another constant. Different pairs of constants are used for each line of coefficients. The encoded file is then read back by the computer. The file is decoded, and the decoded data is displayed on the computer screen. This gives the instructor the opportunity to verify the contents of STU.DAT. The instructor may also receive a printed copy of the surface data. Each time the instructor executes this program, STU.DAT is erased and then reconstructed.

RSM.EXE

RSM.EXE is the primary program in the simulation package. This program begins by reading the encoded data from STU.DAT. It then decodes the data and prompts the user for his surface number, the number of independent variables, the number of distinct design points, and the order of the model being used. The program then prompts the user for the number of repetitions at each point as well as the location of each point. The program summarizes this data on the computer screen so that the student may verify the input. If there is an input error, the student must reenter all data. Once this is complete, the program generates a response value for each observation in the experiment. A random number generator is used to calculate the experimental error at each design point. This error is based on the given error variance in the surface data. The error range is restricted to within four standard deviations of the mean (or zero). Next, the program internally rearranges the data in order to make the regression and ANOVA calculations. These calculations are made using the regression and analysis of variance techniques described in the previous chapter. The program produces three blocks of information. First, it prints the rearranged input data. This includes the value of the response variable, design point number, and the values of the independent variables. Second, the program prints a standard ANOVA table which includes the regression coefficients and the necessary F statistics to conduct the significance of coefficients and goodness of fit tests. Third, the program prints a table with generated responses, the forecasted responses (from the regression equation), their difference, and their differences squared. The sum of the squared differences is also equal to the residual sum of the squares in the ANOVA table. Finally, the program enables the student to obtain a hard copy of the results and to begin another set of experiments.

CRIT.FOR

The final program is executed after an adequate second-order model has been found. The student is prompted to input the coefficients of his second-order model. The student then verifies the equation by reviewing it on the computer screen. The program calculates the first partial derivative with respect to each independent variable and sets them equal to zero. These equations are then solved by matrix algebra techniques. The program displays and prints the value of the independent variables and the response variable at the optimal point. The student then has the option to let the program generate data to map the response contours. The student inputs the y value of the contour, as well as the maximum, minimum, and incremental values for each independent variable. The increment (delta) will dictate how many y values are generated. For example, assume that the minimum and maximum values for X_1 are 1.0 and 2.0 respectively. If the delta value is .2 for X_1, then responses will be generated for X_1 equal to 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. The program calculates the response variable at all points (incremented by delta) between the stated ranges of the independent variables. If the response value is within .01 units of the interested contour, the values for the response variable and the independent variables are displayed. The student can increase the size of the output by decreasing delta or extending the range between the minimum and maximum values for each variable of interest. Trial and error may be necessary when specifying the parameters used to calculate the responses in order to get a reasonable amount of data points to plot the contours. The program allows the student to continue to plot as many contours as he desires.

V.   INSTRUCTOR MANUAL

RSSP- A FORTRAN SIMULATION PACKAGE

FOR USE IN TEACHING RESPONSE

SURFACE METHODOLOGY

INSTRUCTOR MANUAL

OVERVIEW

This manual explains how to use the Response Surface Simulation Package (RSSP). This package is used in the instruction of Response Surface Methodology (RSM). It was developed by James T. Treharne, an Industrial Engineering graduate student at Auburn University. The programs operate on an IBM (or compatible) personal computer. The simulation is a complete revision of a similar set of programs called Response Surface Analysis Program (RSAP). Jesse Martin, a former Auburn University graduate student, developed RSAP in the early 1970's. RSSP enables an instructor to test a student's knowledge of Response Surface Methodology by giving him a simple, yet realistic RSM problem to solve. Additionally, an instructor can use RSSP to generate examples to reinforce the theoretical concepts taught in the classroom.

RSSP is designed for use in graduate level engineering courses that teach RSM. RSM techniques are thoroughly discussed by Montgomery [2]. A student should have an understanding of multiple regression, experimental design, and analysis of variance. The package has sufficient capabilities and flexibility to meet most teaching needs. RSSP allows the instructor to load the equations of at most 15 unique surfaces into the computer. The student is then required, without knowledge of the true surface equation, to use proper RSM techniques to arrive near the optimum region, where the assigned surface is estimated by a second-order model. The student can also find the estimated optimal point and obtain data to plot estimated response contours. The simulation package has the following capabilities:

1. The simulation package is designed to accommodate at most five independent variables. It is recommended that most introductory courses in RSM, at least initially, use only two independent variables.
2. The instructor may provide data for at most fifteen different response surfaces. This normally enables each student to experiment with a unique surface.
3. The experimental error is assumed normally distributed with a mean equal to zero. The instructor must specify the size of the error variance.
4. The maximum number of observations is restricted to sixty. That is to say, the degrees of freedom for the corrected total SS's cannot exceed 59. This number is sufficient for almost any practical situation because an experimenter should always design an experiment that minimizes cost while providing the necessary amount of degrees of

freedom to estimate the coefficients and error variance.

5. Each surface that the instructor inputs is represented by a second-order equation, where some of the coefficients may be set to zero. Its general form is

$$Y = \beta_o + \sum_{i=1}^{5} \beta_{ii} X_i^2 + \sum_{i=1}^{4} \sum_{j=i+1}^{5} \beta_{ij} X_i X_j + \sum_{i=1}^{5} \beta_i X_i .$$

6. The student is responsible for preparing a proper experimental design. Failure to do so may provide invalid results. For example, if he uses an unbalanced design, the equations used to calculate the various sum of the squares terms are invalid.

REVIEW OF RSM

The primary goal of RSM is to find the set of conditions which optimize a given response surface. RSM is an iterative approach that finds the optimal set of conditions as quickly and as efficiently as possible. The general steps are:

1. Design a 1st-order experiment. The design must include a sufficient number of observation points to estimate the regression coefficients, the experimental error, and to test for goodness of

fit. The test should also use the proper spacing, which depends heavily on experimental error.

2. Conduct first-order experiments.
3. Determine if a first-order model is adequate. If an adequate fit exists and at least one independent variable is significant, move along the path of steepest ascent (descent) to the next center point. If there is not a good fit, adjust the spacing until a satisfactory fit is obtained. If a good fit is impossible and the coefficients remain insignificant, it is time to try a second-order model. On the other hand, if there is a good fit but no significant coefficients, the experimenter should increase the spacing. It is time to move to a second-order model when the experimenter can achieve a good fit but cannot achieve significant coefficients.
4. Design a second-order experiment. There must be a sufficient number of observations as in the first-order design.
5. Conduct the second-order experiments.
6. From the ANOVA table determine if there is an adequate fit. If there is a good fit and the coefficients are significant, expand the spacing until the F(LOF) is 75% of the critical F value. If there is not a good fit, the experimenter must

decrease the spacing. If the fit is adequate and there are no significant coefficients, the spacing must be increased.

7. Estimate the optimal value for the dependent variable and the values of the independent variables where the optimum occurs.

8. Map contours of the response variable.

An additional source highlights an excellent strategy for progressing through the above steps [1].

RSSP PROGRAMMING

RSSP consists of three executable Fortran programs: RSMSU.EXE, RSM.EXE, and CRIT.EXE. The instructor uses all three of the programs. Meanwhile, the student uses only RSM.EXE and CRIT.EXE. The package also uses two data files. The first data file, INSTR.DAT, contains the data for the surfaces. The student should not be given a copy of this file. The second file, STU.DAT, is created by the instructor with RSMSU.EXE. This data file contains an encoded copy of the surface data. A detailed explanation of the three programs follows.

RSMSU.EXE

This program reads an instructor prepared data file (INSTR.DAT). This file contains all the vital data for at most fifteen response surfaces. The program then uses this data file to create an encoded data file (STU.DAT). The student uses STU.DAT when he executes the main program. This program is used to ensure that the student does not have access to the uncoded surface data in INSTR.DAT. Prior to executing this setup program, the instructor must use a text editor to create an ASCII file called INSTR.DAT. Each surface contains the following six lines of information.

1. Coefficients of the second-order terms, β_{ii}.

   $(\beta_{11}, \beta_{22}, \beta_{33}, \beta_{44}, \beta_{55})$

2. Coefficients of the interaction terms, β_{ij}.

   $(\beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{23}, \beta_{24}, \beta_{25}, \beta_{34}, \beta_{35}, \beta_{45})$

3. Coefficients of the first-order terms, β_{i}.

   $(\beta_{1}, \beta_{2}, \beta_{3}, \beta_{4}, \beta_{5})$

4. Constant term, β_{o}.

5. Experimental error variance, σ_{ϵ}^{2}.

6. Center point for the first set of experiments.

   $(X_1, X_2, X_3, X_4, X_5)$

Each line must contain a value for every coefficient associated with the variables, even if it is zero. Values should be separated by a space or a comma. Do not combine lines together. RSMSU.EXE first reads and stores the data from INSTR.DAT. The program then encodes the data by adding a constant to each coefficient and then multiplying by another constant. Different pairs of constants are used for each line of coefficients. The encoded file is then read back by the computer. The file is decoded, and the decoded data is displayed on the computer screen. This gives the instructor the opportunity to verify the contents of STU.DAT. The instructor may also receive a printed copy of the surface data. Each time the instructor executes this program, STU.DAT is erased and then reconstructed. Table 2 contains a copy of INSTR.DAT that contains data for three surfaces.

```
-2,-47,0,0,0
10,0,0,0,0,0,0,0,0,0
8,14,0,0,0
81
.05
-4,8,0,0,0
-16,-2,0,0,0
8,0,0,0,0,0,0,0,0,0
-14,47,0,0,0
83
.025
-1,11,0,0,0
11,7,0,0,0
-8,0,0,0,0,0,0,0,0,0
-3,-18,0,0,0
-16
.05
8,-4,0,0,0
```

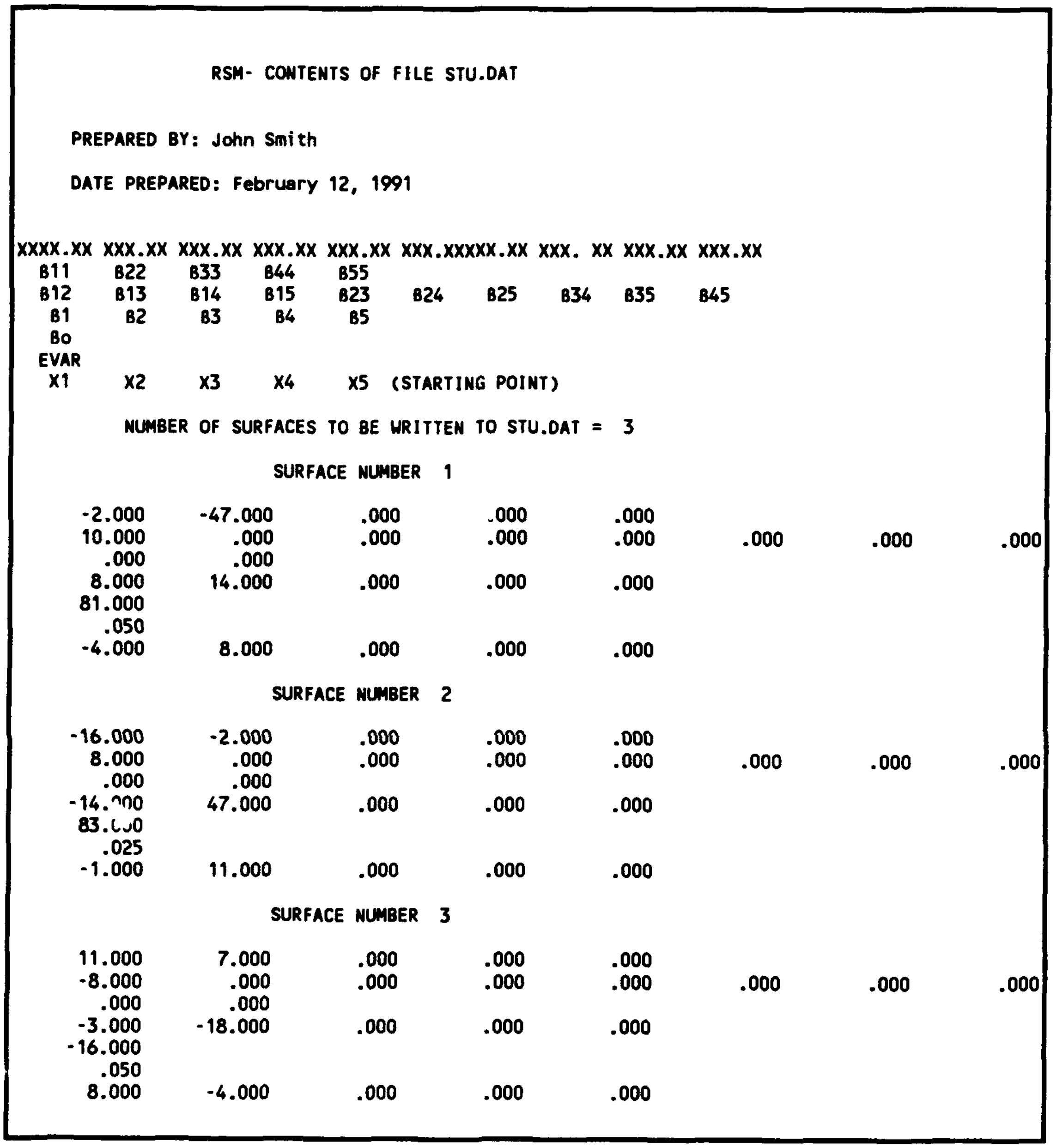
Table 2. File- INSTR.DAT

The equation for surface number one from Table 2 is:

$$Y= -2X_1^2- 47X_2^2+ 10X_1X_2+ 8X_1+ 14X_2+ 81.$$

The experimental error variance is equal to .05. The starting point for the first set of experiments is $X_1= -4$ and $X_2= 8$. The program is executed by typing "RSMSU." The program prompts the instructor for all the necessary information. The following information pertains to the requested input.

1. NAME- Enter up to 25 characters.
2. DATE- Enter in any format up to 25 characters.
3. Number of Surfaces- An error will occur if one inputs a number greater than the number of surfaces in INSTR.DAT. If a number is input that is less than the number of surfaces in INSTR.DAT, the data on the additional surfaces will not be used.
4. The program will display the data on each surface to allow verification by the instructor. If an error is found, it will be necessary to revise INSTR.DAT and execute the program again.
5. The program will generate, if desired, a listing of the surface data which has been written to STU.DAT. Table 3 shows this output after using RSMSU with the data file from Table 2.



```
                    RSM- CONTENTS OF FILE STU.DAT

      PREPARED BY: John Smith

      DATE PREPARED: February 12, 1991

XXXX.XX XXX.XX XXX.XX XXX.XX XXX.XX XXX.XXXXX.XX XXX. XX XXX.XX XXX.XX
  B11     B22    B33    B44    B55
  B12     B13    B14    B15    B23    B24    B25    B34    B35    B45
   B1      B2     B3     B4     B5
   Bo
  EVAR
   X1      X2     X3     X4     X5  (STARTING POINT)

           NUMBER OF SURFACES TO BE WRITTEN TO STU.DAT =  3

                       SURFACE NUMBER  1

     -2.000     -47.000       .000       .000       .000
     10.000        .000       .000       .000       .000       .000       .000       .000
       .000        .000
      8.000      14.000       .000       .000       .000
     81.000
       .050
     -4.000       8.000       .000       .000       .000

                       SURFACE NUMBER  2

    -16.000      -2.000       .000       .000       .000
      8.000        .000       .000       .000       .000       .000       .000       .000
       .000        .000
    -14.000      47.000       .000       .000       .000
     83.000
       .025
     -1.000      11.000       .000       .000       .000

                       SURFACE NUMBER  3

     11.000       7.000       .000       .000       .000
     -8.000        .000       .000       .000       .000       .000       .000       .000
       .000        .000
     -3.000     -18.000       .000       .000       .000
    -16.000
       .050
      8.000      -4.000       .000       .000       .000
```

Table 3. Sample Output from RSMSU.EXE

RSM.EXE

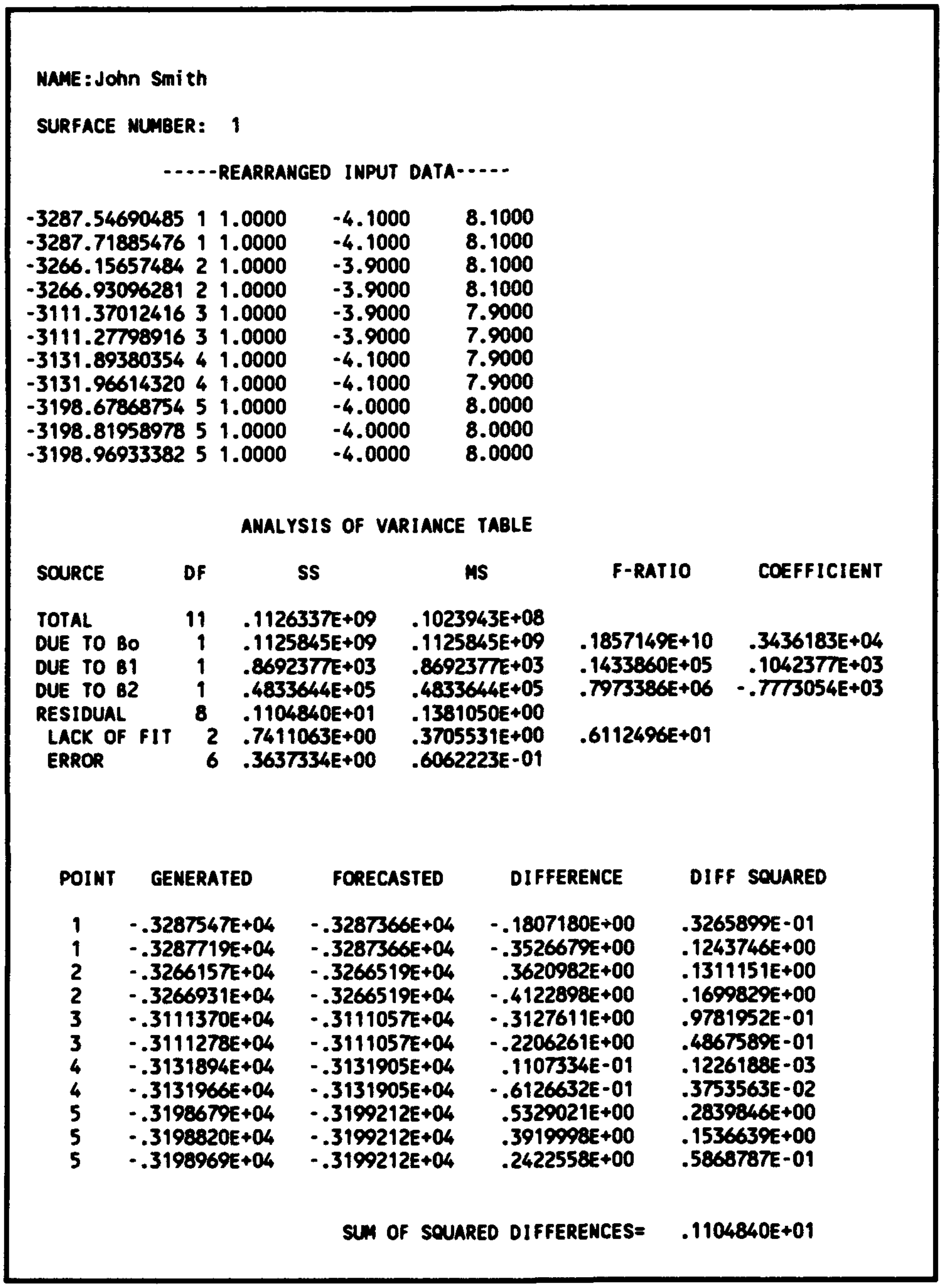
RSM.EXE is the primary program in the simulation package. This program begins by reading the encoded data from STU.DAT. It then decodes the data and prompts the user for the surface number, the number of independent variables, the number of distinct design points, and the order of the model being used. The program then prompts the user for the number of repetitions at each point as well as the location of each point. The program summarizes this data on the computer screen so that the student may verify the input. If there is an input error, the student must reenter the data. Once this is complete, the program generates a response value for each observation in the experiment. A random number generator is used to calculate the experimental error at each design point. This error is based on the given error variance in the surface data. The error range is restricted to within four standard deviations of the mean (or zero). The program produces three blocks of information. First, it prints the rearranged input data. This includes the value of the response variable, design point number, and the values of the independent variables. Second, the program prints a standard ANOVA table which includes the regression coefficients and the necessary F statistics to conduct the significance and goodness of fit tests. Third, the program prints a table with generated responses, the forecasted responses (from the regression equation), their difference, and their differences squared. The sum of the squared differences is also equal to the residual sum of the squares in the ANOVA table. Finally, the program enables the student to obtain a hard copy of the results and to begin another set of experiments. The following points pertain to the execution of this program.

1. NAME- Enter up to 25 characters.
2. Surface Number- Each student is assigned his own surface number.
3. Independent Variables- Maximum of five. Instructor must inform the student of this number.
4. First Design Point- The instructor can include an initial starting point in STU.DAT. This is the center of the initial set of first-order experiments.
5. The program will summarize the data input. If an error was made during input, the data must be reentered. The values are printed with a precision of four decimal places.

An example output is shown in Table 4. This is a first-order experiment. The surface equation used is

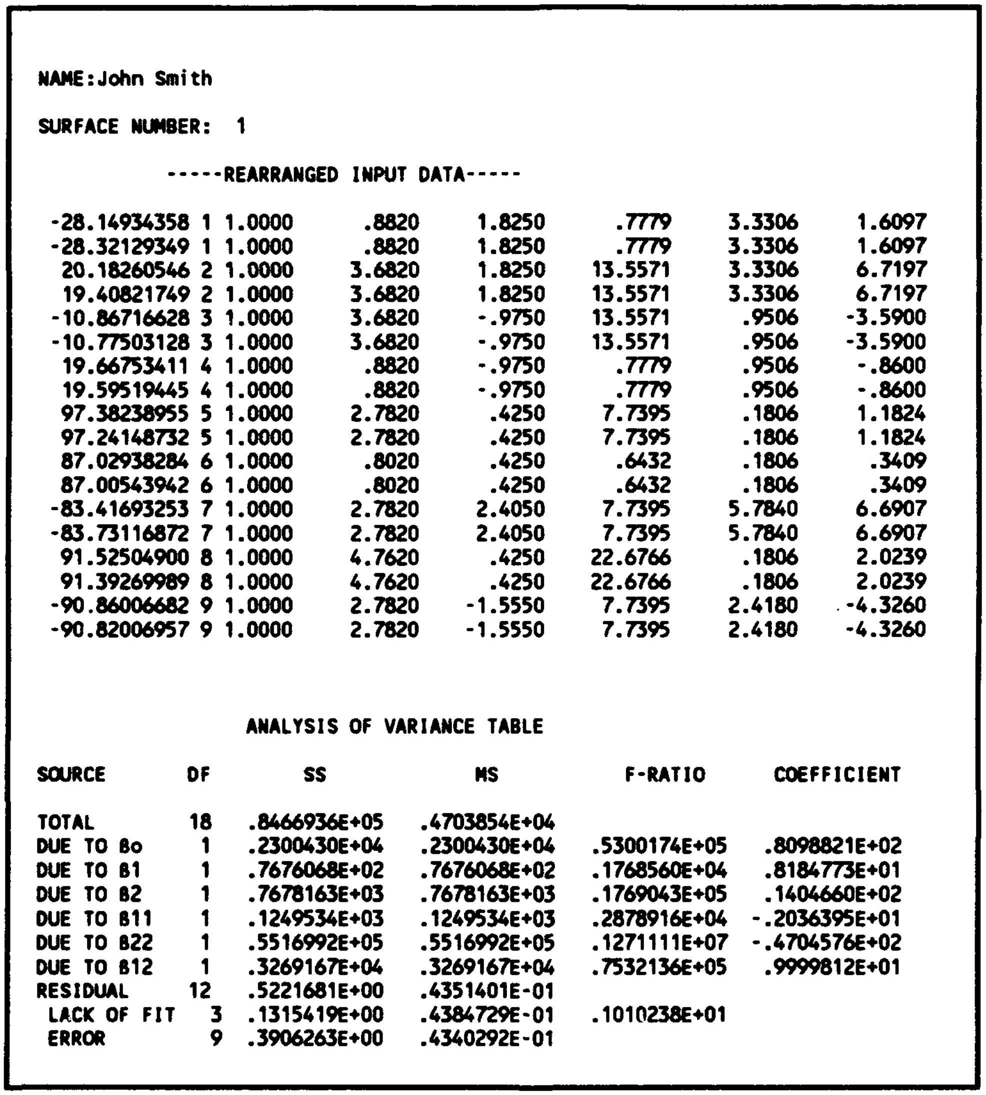
$$Y= -2X_1^2- 47X_2^2+ 10X_1X_2+ 8X_1+ 14X_2 +81.$$

An example output of a second-order experiment using the same surface is shown in Table 5.



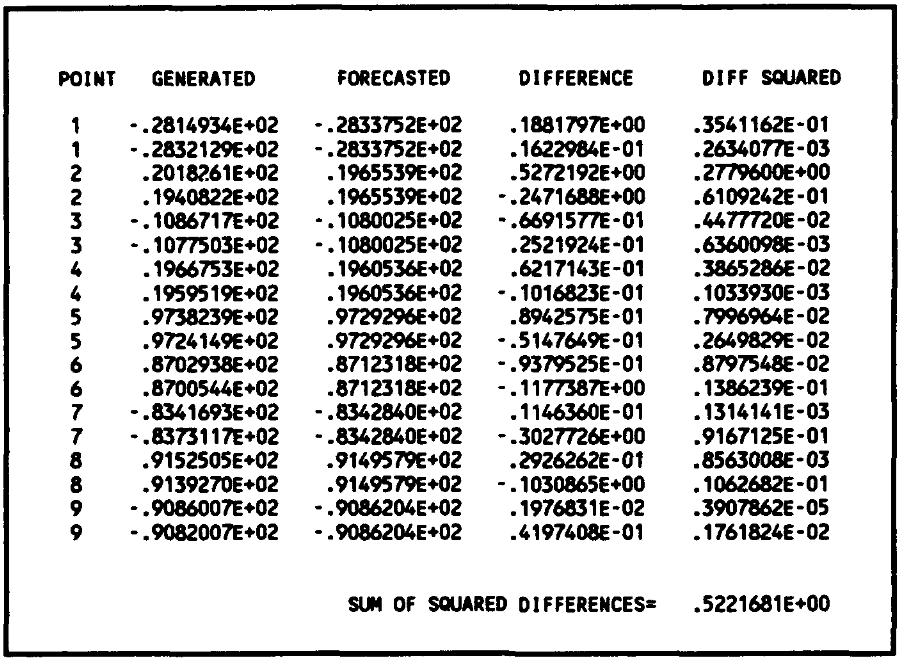
```
NAME:John Smith

SURFACE NUMBER:  1

           -----REARRANGED INPUT DATA-----

-3287.54690485 1 1.0000    -4.1000     8.1000
-3287.71885476 1 1.0000    -4.1000     8.1000
-3266.15657484 2 1.0000    -3.9000     8.1000
-3266.93096281 2 1.0000    -3.9000     8.1000
-3111.37012416 3 1.0000    -3.9000     7.9000
-3111.27798916 3 1.0000    -3.9000     7.9000
-3131.89380354 4 1.0000    -4.1000     7.9000
-3131.96614320 4 1.0000    -4.1000     7.9000
-3198.67868754 5 1.0000    -4.0000     8.0000
-3198.81958978 5 1.0000    -4.0000     8.0000
-3198.96933382 5 1.0000    -4.0000     8.0000


                ANALYSIS OF VARIANCE TABLE

SOURCE        DF       SS             MS              F-RATIO      COEFFICIENT

TOTAL         11   .1126337E+09   .1023943E+08
DUE TO Bo      1   .1125845E+09   .1125845E+09   .1857149E+10   .3436183E+04
DUE TO B1      1   .8692377E+03   .8692377E+03   .1433860E+05   .1042377E+03
DUE TO B2      1   .4833644E+05   .4833644E+05   .7973386E+06  -.7773054E+03
RESIDUAL       8   .1104840E+01   .1381050E+00
 LACK OF FIT   2   .7411063E+00   .3705531E+00   .6112496E+01
 ERROR         6   .3637334E+00   .6062223E-01



  POINT   GENERATED       FORECASTED      DIFFERENCE      DIFF SQUARED

    1   -.3287547E+04   -.3287366E+04   -.1807180E+00   .3265899E-01
    1   -.3287719E+04   -.3287366E+04   -.3526679E+00   .1243746E+00
    2   -.3266157E+04   -.3266519E+04    .3620982E+00   .1311151E+00
    2   -.3266931E+04   -.3266519E+04   -.4122898E+00   .1699829E+00
    3   -.3111370E+04   -.3111057E+04   -.3127611E+00   .9781952E-01
    3   -.3111278E+04   -.3111057E+04   -.2206261E+00   .4867589E-01
    4   -.3131894E+04   -.3131905E+04    .1107334E-01   .1226188E-03
    4   -.3131966E+04   -.3131905E+04   -.6126632E-01   .3753563E-02
    5   -.3198679E+04   -.3199212E+04    .5329021E+00   .2839846E+00
    5   -.3198820E+04   -.3199212E+04    .3919998E+00   .1536639E+00
    5   -.3198969E+04   -.3199212E+04    .2422558E+00   .5868787E-01

                        SUM OF SQUARED DIFFERENCES=   .1104840E+01
```

Table 4. RSM Output (First-Order Model)



```
NAME:John Smith

SURFACE NUMBER:  1

           -----REARRANGED INPUT DATA-----

 -28.14934358 1 1.0000     .8820    1.8250      .7779    3.3306    1.6097
 -28.32129349 1 1.0000     .8820    1.8250      .7779    3.3306    1.6097
  20.18260546 2 1.0000    3.6820    1.8250    13.5571    3.3306    6.7197
  19.40821749 2 1.0000    3.6820    1.8250    13.5571    3.3306    6.7197
 -10.86716628 3 1.0000    3.6820    -.9750    13.5571     .9506   -3.5900
 -10.77503128 3 1.0000    3.6820    -.9750    13.5571     .9506   -3.5900
  19.66753411 4 1.0000     .8820    -.9750      .7779     .9506    -.8600
  19.59519445 4 1.0000     .8820    -.9750      .7779     .9506    -.8600
  97.38238955 5 1.0000    2.7820     .4250     7.7395     .1806    1.1824
  97.24148732 5 1.0000    2.7820     .4250     7.7395     .1806    1.1824
  87.02938284 6 1.0000     .8020     .4250      .6432     .1806     .3409
  87.00543942 6 1.0000     .8020     .4250      .6432     .1806     .3409
 -83.41693253 7 1.0000    2.7820    2.4050     7.7395    5.7840    6.6907
 -83.73116872 7 1.0000    2.7820    2.4050     7.7395    5.7840    6.6907
  91.52504900 8 1.0000    4.7620     .4250    22.6766     .1806    2.0239
  91.39269989 8 1.0000    4.7620     .4250    22.6766     .1806    2.0239
 -90.86006682 9 1.0000    2.7820   -1.5550     7.7395    2.4180   -4.3260
 -90.82006957 9 1.0000    2.7820   -1.5550     7.7395    2.4180   -4.3260


                  ANALYSIS OF VARIANCE TABLE

SOURCE        DF        SS              MS              F-RATIO       COEFFICIENT

TOTAL         18   .8466936E+05   .4703854E+04
DUE TO βo      1   .2300430E+04   .2300430E+04   .5300174E+05   .8098821E+02
DUE TO β1      1   .7676068E+02   .7676068E+02   .1768560E+04   .8184773E+01
DUE TO β2      1   .7678163E+03   .7678163E+03   .1769043E+05   .1404660E+02
DUE TO β11     1   .1249534E+03   .1249534E+03   .2878916E+04  -.2036395E+01
DUE TO β22     1   .5516992E+05   .5516992E+05   .1271111E+07  -.4704576E+02
DUE TO β12     1   .3269167E+04   .3269167E+04   .7532136E+05   .9999812E+01
RESIDUAL      12   .5221681E+00   .4351401E-01
 LACK OF FIT   3   .1315419E+00   .4384729E-01   .1010238E+01
 ERROR         9   .3906263E+00   .4340292E-01
```

Table 5. RSM Output (Second-Order Model)

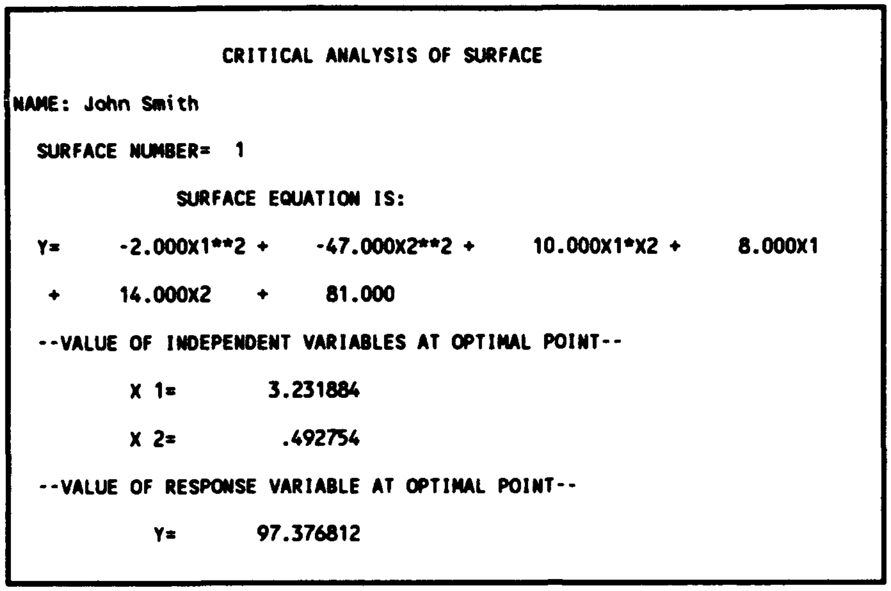


```
POINT   GENERATED       FORECASTED      DIFFERENCE      DIFF SQUARED

  1   -.2814934E+02   -.2833752E+02    .1881797E+00    .3541162E-01
  1   -.2832129E+02   -.2833752E+02    .1622984E-01    .2634077E-03
  2    .2018261E+02    .1965539E+02    .5272192E+00    .2779600E+00
  2    .1940822E+02    .1965539E+02   -.2471688E+00    .6109242E-01
  3   -.1086717E+02   -.1080025E+02   -.6691577E-01    .4477720E-02
  3   -.1077503E+02   -.1080025E+02    .2521924E-01    .6360098E-03
  4    .1966753E+02    .1960536E+02    .6217143E-01    .3865286E-02
  4    .1959519E+02    .1960536E+02   -.1016823E-01    .1033930E-03
  5    .9738239E+02    .9729296E+02    .8942575E-01    .7996964E-02
  5    .9724149E+02    .9729296E+02   -.5147649E-01    .2649829E-02
  6    .8702938E+02    .8712318E+02   -.9379525E-01    .8797548E-02
  6    .8700544E+02    .8712318E+02   -.1177387E+00    .1386239E-01
  7   -.8341693E+02   -.8342840E+02    .1146360E-01    .1314141E-03
  7   -.8373117E+02   -.8342840E+02   -.3027726E+00    .9167125E-01
  8    .9152505E+02    .9149579E+02    .2926262E-01    .8563008E-03
  8    .9139270E+02    .9149579E+02   -.1030865E+00    .1062682E-01
  9   -.9086007E+02   -.9086204E+02    .1976831E-02    .3907862E-05
  9   -.9082007E+02   -.9086204E+02    .4197408E-01    .1761824E-02


                        SUM OF SQUARED DIFFERENCES=    .5221681E+00
```

Table 5 (Continued). RSM Output (Second-order Model)

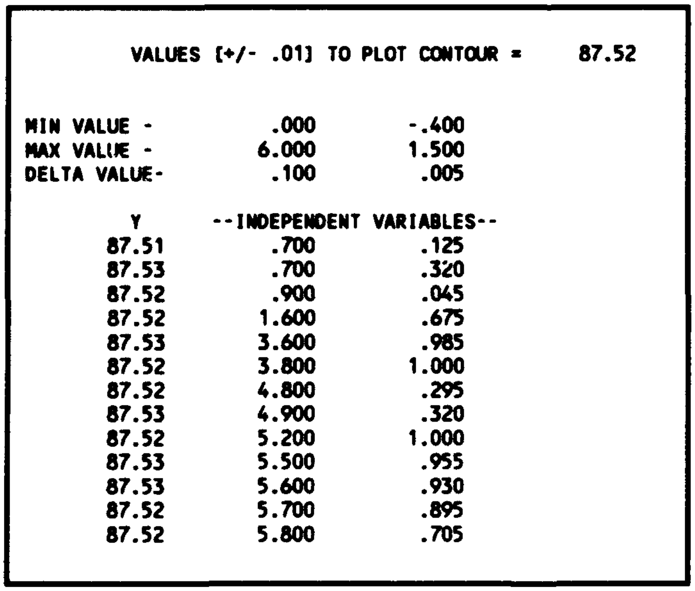
CRIT.EXE

The final program is executed after an adequate second-order model has been found. The student is prompted to input the coefficients of his second-order model. The student then verifies the equation by reviewing it on the computer screen. The program calculates the first partial derivative with respect to each independent variable and sets them equal to zero. These equations are then solved simultaneously by matrix algebra techniques. The program displays and prints the value of the independent variables and the response variable at the optimal point. The student then has the option to let the program compute data to map response contours. The student inputs the y value of the contour, as well as the maximum and minimum values for each independent variable. The student must also input an increment for each independent variable called delta. The delta values will dictate how many response values are generated. For example, assume that the minimum and maximum values for X_1 are 1.0 and 2.0 respectively. If the delta value is .2 for X_1, then responses will be generated for X_1 equal to 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. The program calculates the response variable at all points (incremented by delta) between the stated ranges of the independent variables. If the response value is within .01 units of the interested contour, the values for the response variable and the independent variables are displayed. The student can increase the size of the output by decreasing the delta or extending the range between the minimum and maximum values for each variable of interest. Trial and error may be necessary when specifying the parameters used to generate the responses in order to get a reasonable amount of data points to plot the contours. The program allows the student to continue to plot as many contours as he desires. Table 6 shows the output from CRIT.EXE. This is the optimal point for the same surface as before. Table 7 shows the contour data output from CRIT.EXE.



```
                    CRITICAL ANALYSIS OF SURFACE

NAME: John Smith

  SURFACE NUMBER=  1

              SURFACE EQUATION IS:

  Y=     -2.000X1**2 +    -47.000X2**2 +     10.000X1*X2 +      8.000X1

   +     14.000X2    +     81.000

  --VALUE OF INDEPENDENT VARIABLES AT OPTIMAL POINT--

        X 1=        3.231884

        X 2=         .492754

  --VALUE OF RESPONSE VARIABLE AT OPTIMAL POINT--

          Y=       97.376812
```

Table 6. CRIT Output (Optimal Value)



```
          VALUES [+/- .01] TO PLOT CONTOUR =     87.52


MIN VALUE -          .000        -.400
MAX VALUE -         6.000        1.500
DELTA VALUE-         .100         .005

        Y       --INDEPENDENT VARIABLES--
      87.51          .700         .125
      87.53          .700         .320
      87.52          .900         .045
      87.52         1.600         .675
      87.53         3.600         .985
      87.52         3.800        1.000
      87.52         4.800         .295
      87.53         4.900         .320
      87.52         5.200        1.000
      87.53         5.500         .955
      87.53         5.600         .930
      87.52         5.700         .895
      87.52         5.800         .705
```

Table 7. CRIT Output (Contour Data)

BIBLIOGRAPHY

[1] Maghsoodloo, S. and J.N. Hool, "On Response-Surface Methodology and its Computer Assisted Teaching," *The American Statistician*, Vol 30, pp140-144(1976).

[2] Montgomery, D.C., *Design and Analysis of Experiments*, Wiley, New York (1991).

VI. STUDENT MANUAL

RSSP- A FORTRAN SIMULATION PACKAGE FOR USE IN TEACHING RESPONSE SURFACE METHODOLOGY

STUDENT MANUAL

OVERVIEW

This manual explains to the student how to use the Response Surface Simulation Package (RSSP). This package is used in the instruction of Response Surface Methodology (RSM). It was developed by James T. Treharne, an Industrial Engineering graduate student at Auburn University. The programs operate on an IBM (or compatible) personal computer. The simulation is a complete revision of a similar set of programs called Response Surface Analysis Program (RSAP). Jesse Martin, a former Auburn University graduate student, developed RSAP in the early 1970's. RSSP enables a student to reinforce his knowledge of Response Surface Methodology by giving him a simple, yet realistic RSM problem to solve.

RSSP is designed for use in graduate level engineering courses that teach RSM. RSM techniques are thoroughly discussed by Montgomery [2]. A student should have an understanding of multiple regression, experimental design, and analysis of variance. RSSP allows the instructor to load the equations for at most 15 unique surfaces into the computer. The student is then required, without knowledge of the true surface equation, to use proper RSM techniques to arrive near the optimum region, where the assigned surface is estimated by a second-order model. The student can also find the estimated optimal point and obtain data to plot estimated response contours. The simulation package has the following capabilities:

1. The simulation package is designed to accommodate at most five independent variables. The instructor will inform the student of the number of variables.
2. The instructor may provide data for at most fifteen different response surfaces. This normally enables each student to experiment with a unique surface.
3. The experimental error is assumed normally distributed with a mean equal to zero. The instructor specifies the size of the error variance. The student can estimate the size of the error variance during experimentation.
4. The maximum number of observations is restricted to sixty. That is to say, the degrees of freedom for the corrected total SS's cannot exceed 59. This number is sufficient for almost any practical situation because an experimenter should always design an experiment that minimizes cost while providing the necessary amount of degrees of freedom to estimate the coefficients and error variance.
5. Each surface that the instructor inputs is represented by a second-order equation, where

coefficients may be set to zero. Its general form is

$$Y = \beta_o + \sum_{i=1}^{5} \beta_{ii}X_i^2 + \sum_{i=1}^{4}\sum_{j=i+1}^{5} \beta_{ij}X_iX_j + \sum_{i=1}^{5} \beta_i X_i .$$

6. The student is responsible for preparing a proper experimental design. Failure to do so may provide invalid results. For example, if the student uses an unbalanced design, the equations used to calculate the various sum of the squares terms are invalid.

REVIEW OF RSM

The primary goal of RSM is to find the set of conditions which optimize a given response surface. RSM is an iterative approach that finds the optimal set of conditions as quickly and as efficiently as possible. The general steps are:

1. Design a 1st-order experiment. The design must include a sufficient number of observation points to estimate the regression coefficients, the experimental error, and to test for goodness of fit. The test should also use the proper spacing, which depends heavily on experimental error.
2. Conduct first-order experiments.

3. Determine if a first-order model is adequate. If an adequate fit exists and at least one independent variable is significant, move along the path of steepest ascent (descent) to the next center point. If there is not a good fit, adjust the spacing until a satisfactory fit is obtained. If a good fit is impossible and the coefficients remain insignificant, it is time to try a second-order model. On the other hand, if there is a good fit but no significant coefficients, the experimenter should increase the spacing. It is time to move to a second-order model when the experimenter can achieve a good fit but cannot achieve significant coefficients.
4. Design second-order experiments. There must be a sufficient number of observations as in the first-order design.
5. Conduct the second-order experiments.
6. From the ANOVA table determine if there is an adequate fit. If there is a good fit and the coefficients are significant, expand the spacing until the F(LOF) is 75% of the critical F value. If there is not a good fit, the experimenter must decrease spacing. If the fit is adequate and there are no significant coefficients, the spacing must be increased.

7. Estimate the optimal value for the dependent variable and the values of the independent variables where the optimum occurs.

8. Map contours of the response variable.

An additional source highlights an excellent strategy for progressing through the above steps [1].

RSSP PROGRAMMING

When using RSSP, the student must possess two executable Fortran programs: RSM.EXE and CRIT.EXE. The student must also have a data file, STU.DAT, which contains encoded data for the surfaces. The student should not attempt to make any changes to this data file. Additionally, there is no value to the student to read this data file since it contains encoded data for use by the main program, RSM.EXE. A detailed explanation of the two programs follows.

RSM.EXE

RSM.EXE is the primary program in the simulation package. This program begins by reading the encoded data from STU.DAT. It then decodes the data and prompts the student for his surface number, the number of independent variables, the number of distinct design points, and the order of the model being used. The program then prompts the student for the number of repetitions at each point as well as the location of each point. The program summarizes this data on the computer screen so that the student may verify the input. If there is an input error, the student must reenter the data. Cnce this is complete, the program generates a response value for each observation in the experiment. A random number generator is used to calculate the experimental error at each design point. This error is based on the given error variance in the surface data. The error range is restricted to within four standard deviations of the mean (or zero). The program produces three blocks of information. First, it prints the rearranged input data. This includes the value of the response variable, design point number, and the values of the independent variables. Second, the program prints a standard ANOVA table which includes the regression coefficients and the necessary F statistics to conduct the significance and goodness-of-fit tests. Third, the program prints a table with generated responses, the forecasted responses (from the regression equation), their difference, and their differences squared. The sum of the squared differences is also equal to the residual sum of the squares in the ANOVA table. Finally, the program enables the student to obtain a hard copy of the results and to begin another set of experiments. The following points pertain to the execution of this program.

1. NAME- Enter up to 25 characters.

2. Surface Numper- Each student is assigned his own surface number.

3. Independent Variables- Maximum of five. Instructor must inform the student of this number.

4. First Design Point- The instructor may have included an initial starting point in STU.DAT. This is the center of the initial set of first-order experiments.

5. The program will summarize the data input. If an error was made during input, the data must be reentered. The values are printed with a precision of four decimal places.

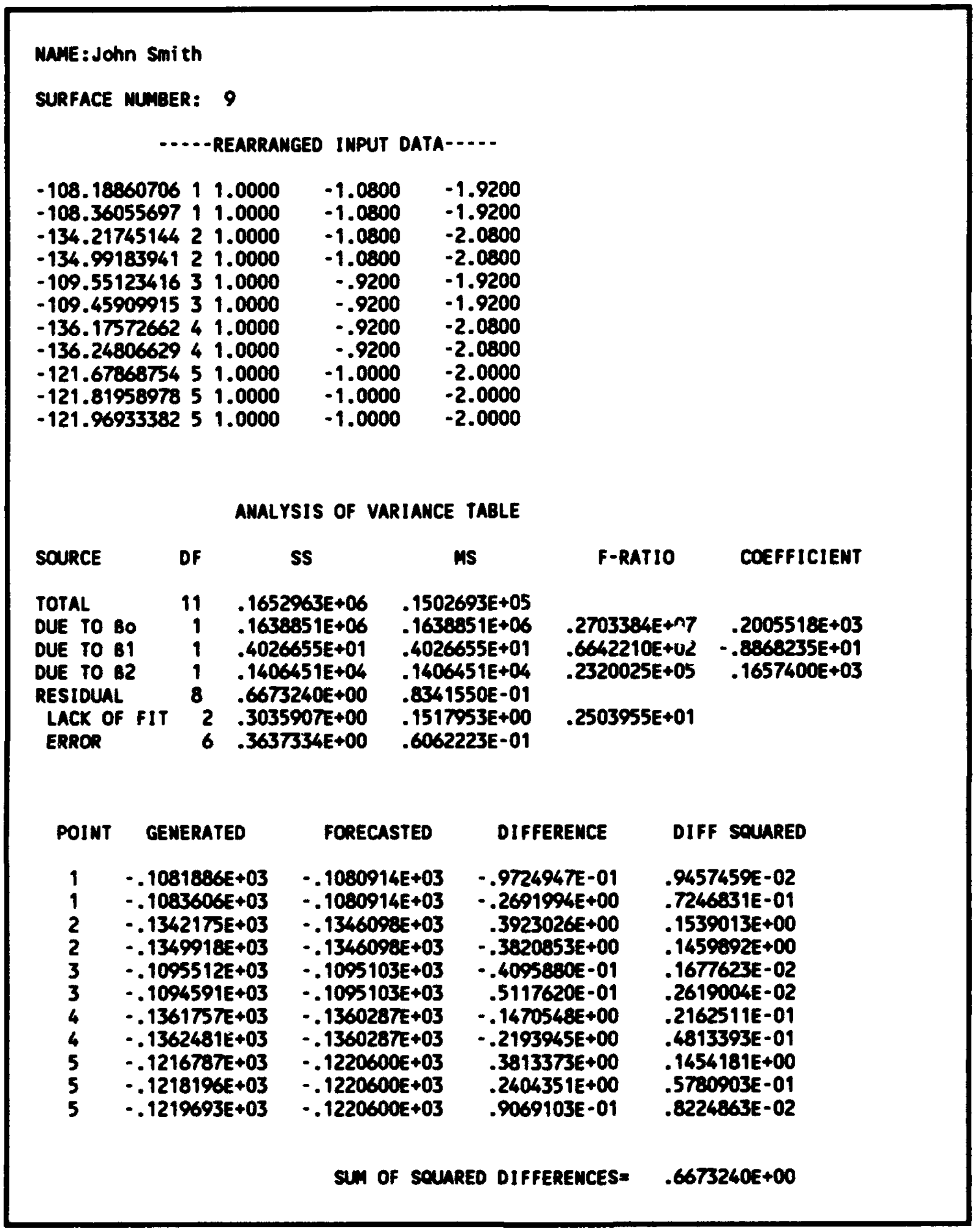
An example output is shown in Table 8. This is a first-order experiment. The exact surface equation used is

$$Y= -4X_1^2- 40X_2^2+ 11X_1X_2+ 6X_1+ 17X_2 +60.$$

The estimated surface equation (for a first-order experiment) is

$$\hat{Y}= -8.868235X_1+ 165.7400X_2+ 200.5518.$$

The F-Ratios show that the three coefficients are all highly significant. Additionally, the lack of fit is insignificant. Therefore, the student should proceed along the path of steepest ascent to the next center point.

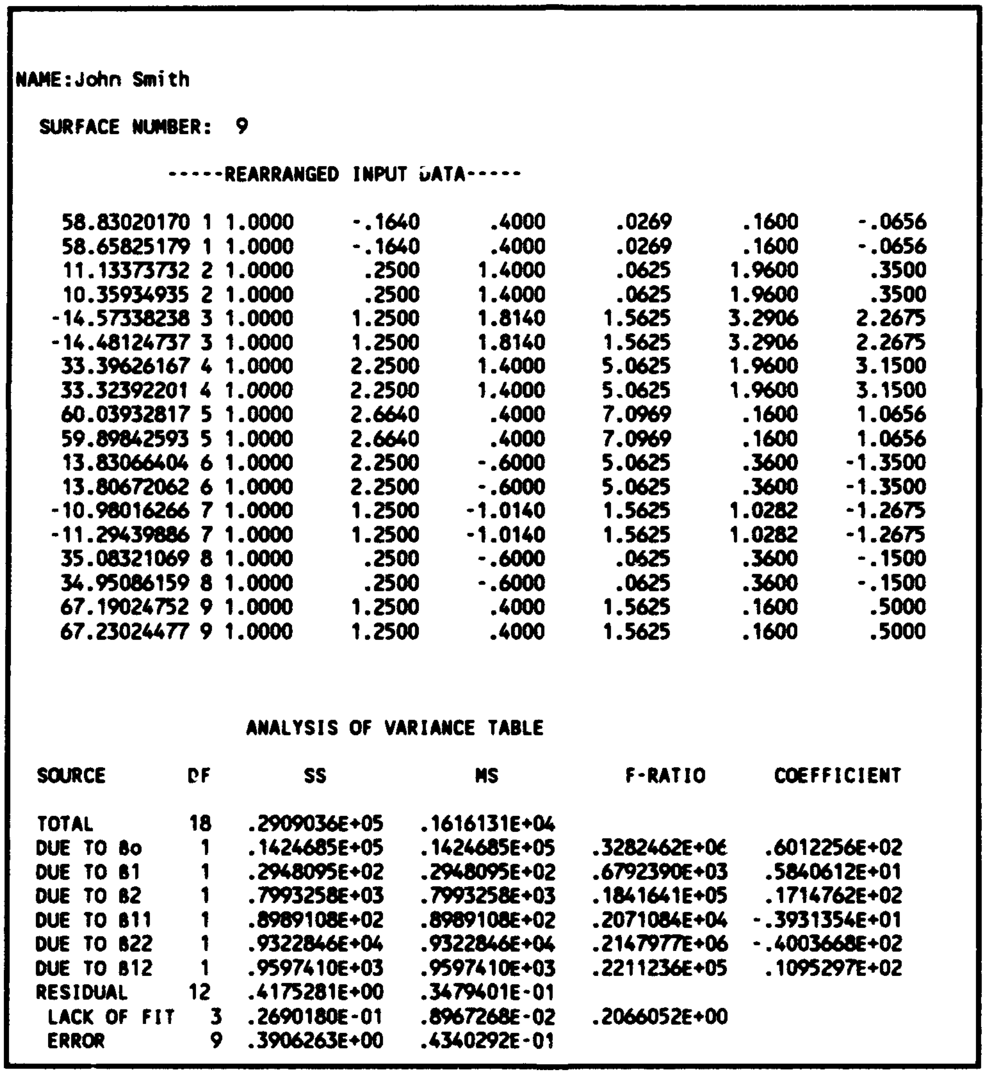


```
NAME:John Smith

SURFACE NUMBER:  9

          -----REARRANGED INPUT DATA-----

-108.18860706 1 1.0000    -1.0800    -1.9200
-108.36055697 1 1.0000    -1.0800    -1.9200
-134.21745144 2 1.0000    -1.0800    -2.0800
-134.99183941 2 1.0000    -1.0800    -2.0800
-109.55123416 3 1.0000     -.9200    -1.9200
-109.45909915 3 1.0000     -.9200    -1.9200
-136.17572662 4 1.0000     -.9200    -2.0800
-136.24806629 4 1.0000     -.9200    -2.0800
-121.67868754 5 1.0000    -1.0000    -2.0000
-121.81958978 5 1.0000    -1.0000    -2.0000
-121.96933382 5 1.0000    -1.0000    -2.0000


                 ANALYSIS OF VARIANCE TABLE

SOURCE       DF       SS            MS             F-RATIO       COEFFICIENT

TOTAL        11   .1652963E+06  .1502693E+05
DUE TO Bo     1   .1638851E+06  .1638851E+06   .2703384E+07   .2005518E+03
DUE TO B1     1   .4026655E+01  .4026655E+01   .6642210E+02  -.8868235E+01
DUE TO B2     1   .1406451E+04  .1406451E+04   .2320025E+05   .1657400E+03
RESIDUAL      8   .6673240E+00  .8341550E-01
 LACK OF FIT  2   .3035907E+00  .1517953E+00   .2503955E+01
 ERROR        6   .3637334E+00  .6062223E-01


  POINT   GENERATED       FORECASTED       DIFFERENCE      DIFF SQUARED

    1    -.1081886E+03   -.1080914E+03   -.9724947E-01    .9457459E-02
    1    -.1083606E+03   -.1080914E+03   -.2691994E+00    .7246831E-01
    2    -.1342175E+03   -.1346098E+03    .3923026E+00    .1539013E+00
    2    -.1349918E+03   -.1346098E+03   -.3820853E+00    .1459892E+00
    3    -.1095512E+03   -.1095103E+03   -.4095880E-01    .1677623E-02
    3    -.1094591E+03   -.1095103E+03    .5117620E-01    .2619004E-02
    4    -.1361757E+03   -.1360287E+03   -.1470548E+00    .2162511E-01
    4    -.1362481E+03   -.1360287E+03   -.2193945E+00    .4813393E-01
    5    -.1216787E+03   -.1220600E+03    .3813373E+00    .1454181E+00
    5    -.1218196E+03   -.1220600E+03    .2404351E+00    .5780903E-01
    5    -.1219693E+03   -.1220600E+03    .9069103E-01    .8224863E-02

                         SUM OF SQUARED DIFFERENCES=   .6673240E+00
```

Table 8. RSM Output (First-Order Model)

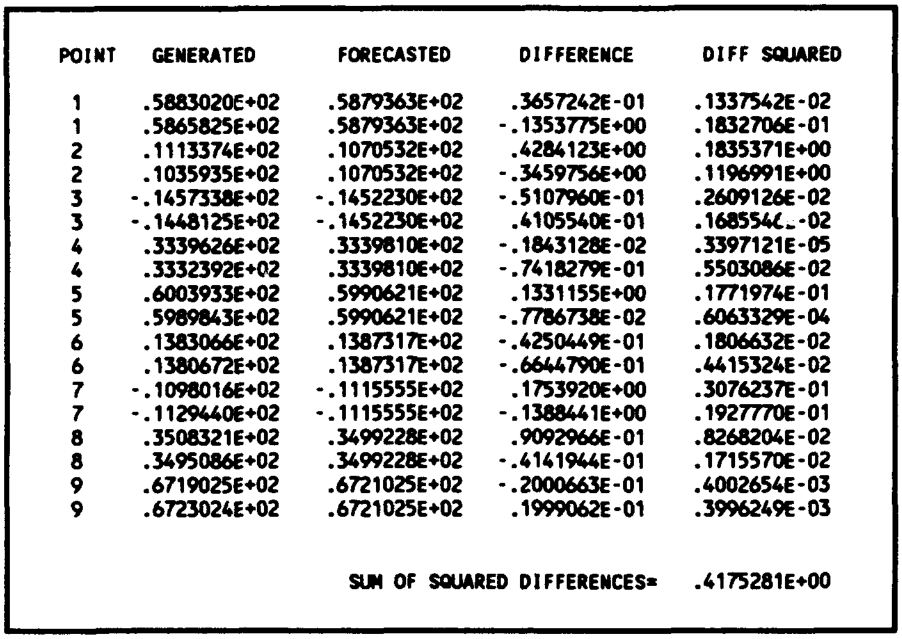
An example output of a second-order experiment using the same surface is shown in Table 9. In this case, the estimated surface equation is

$\hat{Y}= -3.93X_1{}^2- 40.04X_2{}^2+ 10.95X_1X_2+ 5.84X_1+ 17.15X_2+ 60.12.$ All of the coefficients are significant. The lack of fit test indicates a very good fit. The next step is for the student to keep the same center point and expand the spacing until either there is no longer a good fit or one of the coefficients is no longer significant.



```
NAME:John Smith

  SURFACE NUMBER:  9

            -----REARRANGED INPUT DATA-----

    58.83020170 1 1.0000     -.1640      .4000      .0269      .1600     -.0656
    58.65825179 1 1.0000     -.1640      .4000      .0269      .1600     -.0656
    11.13373732 2 1.0000      .2500     1.4000      .0625     1.9600      .3500
    10.35934935 2 1.0000      .2500     1.4000      .0625     1.9600      .3500
   -14.57338238 3 1.0000     1.2500     1.8140     1.5625     3.2906     2.2675
   -14.48124737 3 1.0000     1.2500     1.8140     1.5625     3.2906     2.2675
    33.39626167 4 1.0000     2.2500     1.4000     5.0625     1.9600     3.1500
    33.32392201 4 1.0000     2.2500     1.4000     5.0625     1.9600     3.1500
    60.03932817 5 1.0000     2.6640      .4000     7.0969      .1600     1.0656
    59.89842593 5 1.0000     2.6640      .4000     7.0969      .1600     1.0656
    13.83066404 6 1.0000     2.2500     -.6000     5.0625      .3600    -1.3500
    13.80672062 6 1.0000     2.2500     -.6000     5.0625      .3600    -1.3500
   -10.98016266 7 1.0000     1.2500    -1.0140     1.5625     1.0282    -1.2675
   -11.29439886 7 1.0000     1.2500    -1.0140     1.5625     1.0282    -1.2675
    35.08321069 8 1.0000      .2500     -.6000      .0625      .3600     -.1500
    34.95086159 8 1.0000      .2500     -.6000      .0625      .3600     -.1500
    67.19024752 9 1.0000     1.2500      .4000     1.5625      .1600      .5000
    67.23024477 9 1.0000     1.2500      .4000     1.5625      .1600      .5000


                    ANALYSIS OF VARIANCE TABLE

  SOURCE         DF       SS             MS              F-RATIO       COEFFICIENT

  TOTAL          18   .2909036E+05   .1616131E+04
  DUE TO ßo       1   .1424685E+05   .1424685E+05   .3282462E+06   .6012256E+02
  DUE TO ß1       1   .2948095E+02   .2948095E+02   .6792390E+03   .5840612E+01
  DUE TO ß2       1   .7993258E+03   .7993258E+03   .1841641E+05   .1714762E+02
  DUE TO ß11      1   .8989108E+02   .8989108E+02   .2071084E+04  -.3931354E+01
  DUE TO ß22      1   .9322846E+04   .9322846E+04   .2147977E+06  -.4003668E+02
  DUE TO ß12      1   .9597410E+03   .9597410E+03   .2211236E+05   .1095297E+02
  RESIDUAL       12   .4175281E+00   .3479401E-01
   LACK OF FIT    3   .2690180E-01   .8967268E-02   .2066052E+00
   ERROR          9   .3906263E+00   .4340292E-01
```

Table 9. RSM Output (Second-Order Model)

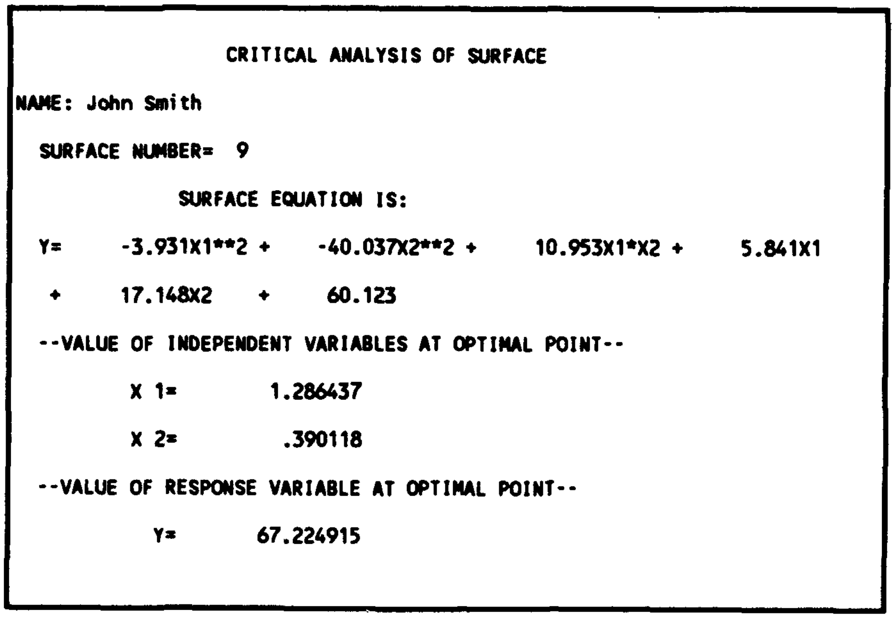


```
POINT   GENERATED       FORECASTED      DIFFERENCE      DIFF SQUARED

  1    .5883020E+02    .5879363E+02    .3657242E-01    .1337542E-02
  1    .5865825E+02    .5879363E+02   -.1353775E+00    .1832706E-01
  2    .1113374E+02    .1070532E+02    .4284123E+00    .1835371E+00
  2    .1035935E+02    .1070532E+02   -.3459756E+00    .1196991E+00
  3   -.1457338E+02   -.1452230E+02   -.5107960E-01    .2609126E-02
  3   -.1448125E+02   -.1452230E+02    .4105540E-01    .1685540E-02
  4    .3339626E+02    .3339810E+02   -.1843128E-02    .3397121E-05
  4    .3332392E+02    .3339810E+02   -.7418279E-01    .5503086E-02
  5    .6003933E+02    .5990621E+02    .1331155E+00    .1771974E-01
  5    .5989843E+02    .5990621E+02   -.7786738E-02    .6063329E-04
  6    .1383066E+02    .1387317E+02   -.4250449E-01    .1806632E-02
  6    .1380672E+02    .1387317E+02   -.6644790E-01    .4415324E-02
  7   -.1098016E+02   -.1115555E+02    .1753920E+00    .3076237E-01
  7   -.1129440E+02   -.1115555E+02   -.1388441E+00    .1927770E-01
  8    .3508321E+02    .3499228E+02    .9092966E-01    .8268204E-02
  8    .3495086E+02    .3499228E+02   -.4141944E-01    .1715570E-02
  9    .6719025E+02    .6721025E+02   -.2000663E-01    .4002654E-03
  9    .6723024E+02    .6721025E+02    .1999062E-01    .3996249E-03

                         SUM OF SQUARED DIFFERENCES=    .4175281E+00
```

Table 9 (Continued). RSM Output (Second-order Model)

CRIT.EXE

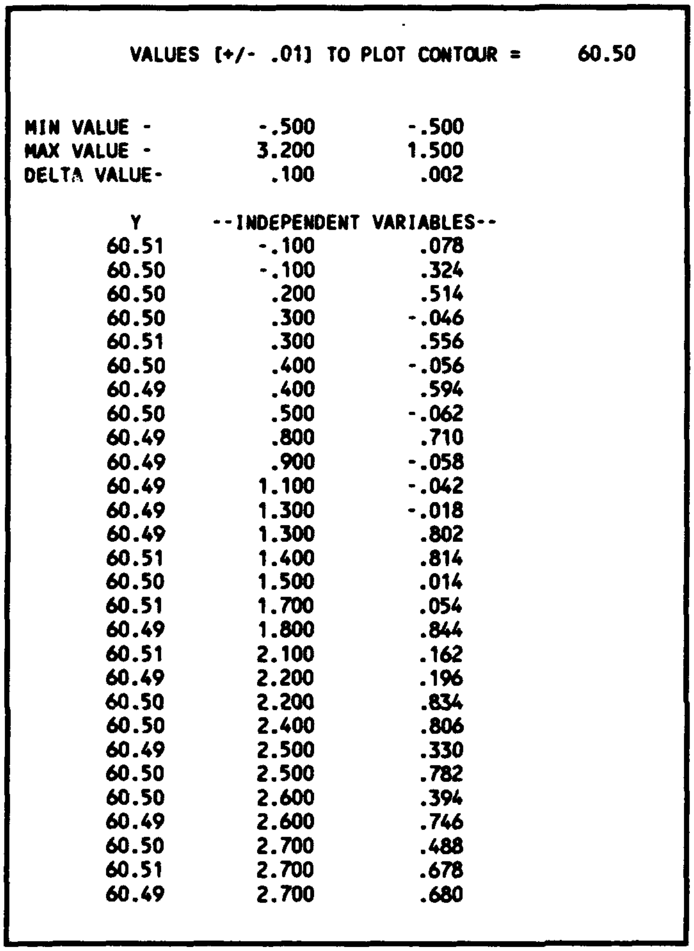
The second program is executed after an adequate second-order model has been found. The student is prompted to input the coefficients of his second-order model. The student then verifies the equation by reviewing it on the computer screen. The program calculates the first partial derivative with respect to each independent variable and sets them equal to zero. These equations are then solved simultaneously by matrix algebra techniques. The program displays and prints the value of the independent variables and the response variable at the optimal point. The student then has the option to let the program generate data to map response contours. The student inputs the y value of the contour, as well as the maximum and minimum values for each independent variable. The student must also input an increment for each variable called delta. The delta values will dictate how many response values are generated. For example, assume that the minimum and maximum values for X_1 are 1.0 and 2.0 respectively. If the delta value is .2 for X_1, then responses will be generated for X_1 equal to 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. The program calculates the response variable at all points (incremented by delta) between the stated ranges of the independent variables. If the response value is within .01 units of the interested contour, the values for the response variable and the independent variables are displayed. The student can increase the size of the output by decreasing delta or extending the range between the minimum and maximum values for each independent variable of interest. Trial and error may be necessary when specifying the parameters used to calculate the responses in order to get a reasonable amount of data points to plot the contours. The program allows the student to continue to plot as many contours as he desires. Table 10 shows the output from CRIT.EXE. This is the optimal point for the same surface as before. The CRIT results at this point are very close to the exact values of the optimal conditions. The exact optimal point is X_1 = 1.285164 and X_2 = .389210. The value of the response variable at the exact optimal point is 67.163776.



```
                  CRITICAL ANALYSIS OF SURFACE
NAME: John Smith
  SURFACE NUMBER=  9
              SURFACE EQUATION IS:
  Y=     -3.931X1**2 +    -40.037X2**2 +     10.953X1*X2 +      5.841X1
   +     17.148X2    +      60.123
  --VALUE OF INDEPENDENT VARIABLES AT OPTIMAL POINT--
         X 1=       1.286437
         X 2=        .390118
  --VALUE OF RESPONSE VARIABLE AT OPTIMAL POINT--
           Y=      67.224915
```

Table 10. CRIT Output (Optimal Value)

Table 11 shows the contour data output from CRIT.EXE.



```
        VALUES [+/- .01] TO PLOT CONTOUR =     60.50


MIN VALUE -         -.500        -.500
MAX VALUE -         3.200        1.500
DELTA VALUE-         .100         .002

      Y        --INDEPENDENT VARIABLES--
    60.51          -.100         .078
    60.50          -.100         .324
    60.50           .200         .514
    60.50           .300        -.046
    60.51           .300         .556
    60.50           .400        -.056
    60.49           .400         .594
    60.50           .500        -.062
    60.49           .800         .710
    60.49           .900        -.058
    60.49          1.100        -.042
    60.49          1.300        -.018
    60.49          1.300         .802
    60.51          1.400         .814
    60.50          1.500         .014
    60.51          1.700         .054
    60.49          1.800         .844
    60.51          2.100         .162
    60.49          2.200         .196
    60.50          2.200         .834
    60.50          2.400         .806
    60.49          2.500         .330
    60.50          2.500         .782
    60.50          2.600         .394
    60.49          2.600         .746
    60.50          2.700         .488
    60.51          2.700         .678
    60.49          2.700         .680
```

Table 11. CRIT Output (Contour Data)

BIBLIOGRAPHY

[1] Maghsoodloo, S. and J.N. Hool, "On Response-Surface Methodology and its Computer Assisted Teaching," *The American Statistician*, Vol 30, pp140-144(1976).

[2] Montgomery, D.C., *Design and Analysis of Experiments*, Wiley, New York (1991).

BIBLIOGRAPHY

[1] Lindfield, G.R. and J.E.T. Penny, *Microcomputers in Numerical Analysis*, Wiley, New York (1989).

[2] Maghsoodloo, S. and J.N. Hool, "On Response-Surface Methodology and its Computer Assisted Teaching," *The American Statistician*, Vol 30, pp140-144 (1976).

[3] Martin, Jesse, "RSAP- A Computerized Simulator For Use In Teaching Response Surface Analysis," Unpublished Thesis, Auburn University (1973).

[4] Montgomery, D.C., *Design and Analysis of Experiments*, Wiley, New York (1991).

APPENDICES

APPENDIX A- RSMSU.EXE PROGRAM LISTING

```
*        ----SETUP PROGRAM-- RESPONSE SURFACE METHODOLOGY ----
*            PROGRAM NAME-- RSMSU.FOR
*
*                                        WRITTEN BY JAMES T. TREHARNE
*                                        MARCH 11, 1991
*
*     THIS PROGRAM IS USED BY THE INSTRUCTOR AS PART OF THE RESPONSE
*     METHODOLOGY SIMULATION.  THIS PROGRAM READS THE INSTRUCTOR'S
*     DATA FILE(INSTR.DAT) WHICH CONTAINS ALL THE REQUIRED DATA ON THE
*     SURFACES.  THE PROGRAM USES THIS DATA FILE TO CREATE AN ENCODED
*     DATA FILE (STU.DAT) WHICH IS GIVEN TO THE STUDENT AND USED BY
*     THE MAIN SIMULATION PROGRAM (RSM.FOR)
*
*

      PROGRAM RSMSU
*
      DOUBLE PRECISION COEF(15,5),CIACT(15,10),CFORD(15,5)
      DOUBLE PRECISION SCOEF(15,5),SCIACT(15,10),SCFORD(15,5)
      DOUBLE PRECISION CONST(15), EVAR(15)
      DOUBLE PRECISION SCONST(15)
      DOUBLE PRECISION START(15,5)
      INTEGER    NSUR, ILIST
      CHARACTER*25 NAME,DATE
*
*                  OPEN FILE CREATED BY INSTRUCTOR
*
      OPEN (10,FILE='INSTR.DAT',STATUS='OLD')
*
*                  OPEN A NEW FILE FOR STUDENT USE
*
      OPEN (11,FILE='STU.DAT',STATUS='NEW')
*
*                  DEFINE PRINTER AS FILE #6
*
      OPEN (6,FILE='PRN',STATUS='NEW')
*
*                  DEFINITION OF VARIABLES
*
*      NSUR- NUMBER OF SURFACES-  MAX IS 15
*      ILIST-  "0"= DO NOT PRINT OPTIONAL REPORT
*              "1"= PRINT OPTIONAL REPORT
*      COEF(15,5)-     COEFFICIENTS OF HIGHER ORDER TERMS,E.G ß11, ß22
*      SCOEF(15,5)-    ENCODED COEFFICIENTS OF HIGHER ORDER TERMS
*      CIACT(15,10)-   COEFFICIENTS OF INTERACTION TERMS, E.G. ß12
*      SCIACT(15,10)-  ENCODED COEFFICIENTS OF INTERACTION TERMS
*      CFORD(15,5)-    COEFFICIENTS OF FIRST ORDER TERMS, E.G. ß1,ß2
*      SCFORD(15,5)-   ENCODED COEFFICIENTS OF FIRST ORDER TERMS
*      CONST(15)-      CONSTANT TERM IN SURFACE EQUATION, ßo
*      SCONST(15)-     ENCODED CONSTANT TERM IN SURFACE EQUATION
*      EVAR(15)-       ERROR VARIANCE USED IN SIMULATION
*      START(15,5)-    STARTING POINT OF 5 VARIABLES
*         -----------------------------------------------------
*                  INITIALIZE VARIABLES
*
      NSUR=0
      ILIST=0
*                  INITIALIZE ALL ARRAYS TO "0"
      DO 15 I=1,15
       CONST(I)=0
       SCONST(I)=0
       EVAR(15)=0
       DO 5 K=1,10
        CIACT(I,K)=0
        SCIACT(I,K)=0
  5    CONTINUE
       DO 10 K=1,5
        COEF(I,K)=0
        SCOEF(I,K)=0
        CFORD(I,K)=0
        SCFORD(I,K)=0
        START(I,K)=0
 10    CONTINUE
 15   CONTINUE
*
*                  SCREEN START UP INFORMATION
*
      WRITE(*,*)
      WRITE(*,20)
 20   FORMAT(1X,'WELCOME TO THE RESPONSE SURFACE METHODOLOGY',
     +' SIMULATION SETUP PROGRAM')
      WRITE(*,*)
      WRITE(*,25)
 25   FORMAT(9X,'This program will create a data file for the',
     +' student',/
     +,9X,  'to use in the simulation.  The data file is called'
     +,/,9X,'STU.DAT.  The data is encoded so it will not aid the'
     +,/,9X,'student if he/she reads it.  You must already have a'
     +,/,9X,'data file named INSTR.DAT which must include a value'
     +,/,9X,'for every term, even if it is zero or not used in the'
     +,/,9X,'model.  The student should be given only the file'
     +,/,9X,'STU.DAT and not INSTR.DAT.  The data should be'
     +,/,9X,'arrayed in the following manner:')
      WRITE(*,*)  ' '
      WRITE(*,30)
 30   FORMAT(1X,'XXXX.XX,XXX.XX,XXX.XX,XXX.XX,XXX.XX,XXX.XX,XXX.XX,'
     +,'XXX.XX,XXX.XX,XXX.XX')
      WRITE(*,*)'  ß11    ß22    ß33    ß44    ß55'
      WRITE(*,*)'  ß12    ß13    ß14    ß15    ß23    ß24',
     +'   ß25   ß34   ß35   ß45'
      WRITE(*,*)'   ß1     ß2     ß3     ß4     ß5'
      WRITE(*,*)'   ßo'
      WRITE(*,*)'  EVAR       '
      WRITE(*,*)'   X1     X2     X3     X4     X5   (STARTING POINT)'
      WRITE(*,*) ' '
      WRITE(*,*) ' '
      CALL CONT
      CALL SKIP(12)
      WRITE(*,*)'                                 PLEASE ENTER YOUR NAME'
      CALL SKIP(12)
      READ(*,32) NAME
      CALL SKIP(12)
 32   FORMAT(A25)
      WRITE(*,*) ' '
      WRITE(*,*)'                                 PLEASE ENTER THE DATE'
      CALL SKIP(12)
      READ(*,33) DATE
      CALL SKIP(12)
 33   FORMAT(A25)
      WRITE(*,*) ' '
 35   WRITE(*,*)'    ENTER THE NUMBER OF SURFACES(1   5) IN YOUR DATA',
     +' SET'
      CALL SKIP(12)
      REWIND(10)
      REWIND(11)
      READ(*,40,ERR=35) NSUR
      CALL SKIP(12)
      WRITE(11,*) NSUR
 40   FORMAT(I2)
 45   FORMAT(1X,10F12.3)
*
*                    READ INSTR.DAT(EACH SURFACE HAS 6 LINES OF DATA)
*
      DO 60 I=1,NSUR
*
        READ(10,*,ERR=140) (COEF(I,J),J=1,5)
        READ(10,*,ERR=140) (CIACT(I,J),J=1,10)
        READ(10,*,ERR=140) (CFORD(I,J),J=1,5)
        READ(10,*,ERR=140) CONST(I)
        READ(10,*,ERR=140) EVAR(I)
        READ(10,*,ERR=140) (START(I,J),J=1,5)
*
*                    ENCODE DATA ON COEFFICIENTS
*
        DO 50 J=1,5
          SCOEF(I,J)= (COEF(I,J)+14)*2
          SCFORD(I,J)= (CFORD(I,J)+21)*3
  50    CONTINUE
        DO 55 J=1,10
          SCIACT(I,J)= (CIACT(I,J)+11)*4
  55    CONTINUE
        SCONST(I)= (CONST(I)+5)*7
*
*                    WRITE TO ENCODED FILE(STU.DAT)
*
        WRITE(11,*) (SCOEF(I,J),J=1,5)
        WRITE(11,*) (SCIACT(I,J),J=1,10)
        WRITE(11,*) (SCFORD(I,J),J=1,5)
        WRITE(11,*) SCONST(I)
        WRITE(11,*) EVAR(I)
        WRITE(11,*) (START(I,J),J=1,5)
  60  CONTINUE
      REWIND (11)
        READ(11,*) NSUR
        CALL SKIP(25)
        WRITE(*,65) NSUR
  65    FORMAT(10X,' NUMBER OF SURFACES TO BE WRITTEN TO STU.DAT ='
     +,I2)
        CALL SKIP(12)
        CALL CONT
        CALL SKIP(25)
  67    DO 85 I=1,NSUR
*
*                  READ BACK ENCODED DATA FILE (STU.DAT)
*
        READ(11,*) (SCOEF(I,J),J=1,5)
        READ(11,*) (SCIACT(I,J),J=1,10)
        READ(11,*) (SCFORD(I,J),J=1,5)
        READ(11,*) SCONST(I)
        READ(11,*) EVAR(I)
        READ(11,*) (START(I,J),J=1,5)
        WRITE(*,*)
        CALL SKIP(20)
*
*                  DECODE DATA FROM STU.DAT
*
        DO 70 J=1,5
          COEF(I,J)= (SCOEF(I,J)/2)-14
          CFORD(I,J)= (SCFORD(I,J)/3)-21
  70    CONTINUE
        DO 75 J=1,10
         CIACT(I,J)= (SCIACT(I,J)/4)-11
  75    CONTINUE
        CONST(I)= (SCONST(I)/7)-5
*
        WRITE(*,*)
*                  WRITE DATA TO SCREEN
        WRITE(*,80)I
  80    FORMAT(25X,'SURFACE NUMBER = ',I2)
*
        WRITE(*,*)
        WRITE(*,45) (COEF(I,J),J=1,5)
        WRITE(*,45) (CIACT(I,J),J=1,10)
        WRITE(*,45) (CFORD(I,J),J=1,5)
        WRITE(*,45) CONST(I)
        WRITE(*,45) EVAR(I)
        WRITE(*,45) (START(I,J),J=1,5)
        CALL SKIP(5)
        CALL CONT
  83    CALL SKIP(25)
  85  CONTINUE
        CALL SKIP(20)
  90  WRITE(*,*)'         DO YOU WANT A PRINTOUT OF THE SURFACE DATA?'
      WRITE(*,*)
      WRITE(*,*)'                   "1" = PRINTOUT'
      WRITE(*,*)'                   "0" = NO PRINTOUT- EXIT PROGRAM'
      CALL SKIP(9)
      READ(*,*,err=130) ILIST
      IF (ILIST.EQ.0) THEN
        GO TO 115
      ENDIF
      IF (ILIST.EQ.1) THEN
        GO TO 95
      ENDIF
      GO TO 90
  95  WRITE(6,*)
      WRITE(6,*)'                  RSM- CONTENTS OF FILE STU.DAT'
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'     PREPARED BY: ',NAME
      WRITE(6,*)
      WRITE(6,*)'     DATE PREPARED: ',DATE
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,100)
 100  FORMAT(1X,'XXXX.XX XXX.XX XXX.XX XXX.XX XXX.XX XXX.XX',
     +' XXX.XX XXX.XX XXX.XX XXX.XX')
      WRITE(*,*)
      WRITE(6,*)'  ß11    ß22    ß33    ß44    ß55'
      WRITE(6,*)'  ß12    ß13    ß14    ß15    ß23    ß24',
     +'    ß25    ß34    ß35    ß45'
      WRITE(6,*)'   ß1     ß2     ß3     ß4     ß5'
      WRITE(6,*)'   ßo'
      WRITE(6,*)'  EVAR     '
      WRITE(6,*)'   X1     X2     X3     X4     X5  (STARTING POINT)'
      WRITE(6,*)
      REWIND (11)
        WRITE(6,65) NSUR
        WRITE(6,*)
      DO 110 I=1,NSUR
*
        WRITE(6,105) I
 105    FORMAT(25X,'SURFACE NUMBER ',I2)
        WRITE(6,*)
*
        WRITE(6,45) (COEF(I,J),J=1,5)
        WRITE(6,45) (CIACT(I,J),J=1,10)
        WRITE(6,45) (CFORD(I,J),J=1,5)
        WRITE(6,45) CONST(I)
        WRITE(6,45) EVAR(I)
        WRITE(6,45) (START(I,J),J=1,5)
        WRITE(6,*)
 110  CONTINUE
 115  WRITE(*,*)'               STU.DAT IS PROPERLY SETUP FOR USE'
      CALL SKIP(12)
 120  STOP
 130  write(*,*)'           YOU MUST ENTER A "1(ONE)" OR "0"(ZERO)'
      WRITE(*,*)
      WRITE(*,*)
      GO TO 90
 140  WRITE(*,*)'ERROR READING INSTR.DAT!  MAKE SURE IT IS FORMATTED'
      WRITE(*,*)'PROPERLY AND HAS DATA FOR ALL SURFACES STATED.'
      GO TO 120
      END
*
*                  SUBROUTINE SKIP-- PRINTS 'N' BLANK LINES
*
      SUBROUTINE SKIP(N)
      DO 150 I=1,N
         WRITE(*,*)' '
 150  CONTINUE
      RETURN
      END
*
*                  SUBROUTINE CONT- HALTS EXECUTION UNTIL USER READY
*
      SUBROUTINE CONT
      CHARACTER*1 ANS,BLK
      DATA BLK/' '/
      ANS=BLK
      WRITE(*,1)
   1  FORMAT(/,'                      To continue, press RETURN key')
      READ(*,2) ANS
   2  FORMAT(A1)
      RETURN
      END
```

APPENDIX B- RSM.EXE PROGRAM LISTING

```
*     ----------------MAIN PROGRAM------------------------------
*                     PROGRAM NAME:  RSM.FOR
*                                    WRITTEN BY JAMES T. TREHARNE
*                                    APRIL 3, 1991
*
*
      PROGRAM RSM
      DIMENSION COEF(15,5),CIACT(15,10),CFORD(15,5),
     +CONST(15),EVAR(15)
      INTEGER NSUR,IORDER,ILIST,NVAR,NSTUD
*
      DIMENSION SCOEF(15,5),SCIACT(15,10),SCFORD(15,5)
      DIMENSION SCONST(15),START(15,5),Z(60,6)
*
      DIMENSION NPT(60),IDEF(25)
      DOUBLE PRECISION Y(60),XT(25,60),X(60,25),A(25,25),AA(25,25),
     +B(25),SS(25),G(60),AINV(25,25),
     +CON,VAL,TEST(25,25),TMS(25),FRATIO(25),SAM,SUM,SSR,TT,ESS,EESS,
     +TEESS,YD(60),YF(60),COUNT,YDTOTAL,YDSQ(60)
*
      REAL*8 SEED
      CHARACTER NAME*25
*
*                  DEFINITION OF VARIABLES
*
*     COEF(15,5)-  COEFFICIENTS OF HIGHER ORDER TERMS,E.G. ß11,ß22
*     SCOEF(15,5)- CODED COEF. OF HIGHER ORDER TERMS,E.G. ß11,ß22
*     CIACT(15,10)-COEFFICIENTS OF INTERACTION TERMS,E.G. ß12
*     SCIACT(15,10)-CODED COEFFICIENTS OF INTERACTION TERMS,E.G. ß12
*     CFORD(15,5)- COEFFICIENTS OF FIRST ORDER TERMS,E.G. ß1
*     SCFORD(15,5)-CODED  COEFFICIENTS OF FIRST ORDER TERMS,E.G. ß1
*     CONST(15)-   CONSTANT TERM IN SURFACE EQUATION
*     SCONST(15)-  CODED CONSTANT TERM IN SURFACE EQUATION
*     EVAR(15)-  ERROR VARIANCE
*     START(15,5)- STARTING POINT OF 5 VARIABLES
*     NSUR- NUMBER OF SURFACES- MAX IS 15
*     NSTUD- STUDENT/SURFACE NUMBER
*     NVAR- # INDEPENDENT VARIABLES
*     NDOBS- # OF DISTINCT POINTS FOR EXPERIMENTATION
*     NOBS-  # OF OBSERVATIONS(TOTAL)
*     IORDER-  USED TO DETERMINE 1ST/2ND ORDER EQUATION
*     NREPS-  # OF OBSERVATIONS AT A GIVEN DISTINCT POINT
*     ILIST- USED TO PRINT HARDCOPY RESULTS
*     NPT(50)- VALUE OF DISTINCT POINT NUMBER FOR UP TO 50 OBS.
*     Z(60,6)- USED TO INPUT VALUES OF DISTINCT OBSERVATIONS
*     Y(60)- RESPONSE VARIABLE AT EACH POINT
*     XT(25,60) TRANSPOSE OF MATRIX X
*     X(60,25)- INDEPENDENT TERMS, DUMMY,X1,X2,X3,X4,X5,X1**2,...
*     A(25,25)- MATRIX XT * X
*     AA(25,25)- COPY OF MATRIX A USED TO GET INVERSE
*     AINV(25,25)- IDENTITY MATRIX USED TO GET INVERSE OF A
*     B(25)- COEFFICIENTS OF FITTED EQUATION
*     YF(60)- FORECASTED RESPONSE
*     YD(60)- DIFFERENCE BETWEEN FORECASTED AND GENERATED RESPONSE
*     SS(25)- SUM OF SQUARES TERMS
*     IDEF(25)- DEGREE OF FREEDOM TERMS
*     TMS(25)- MEAN SQUARE VALUES
*     FRATIO(25)- VALUE FROM F-TEST
*     YDTOTAL-SUM OF SQUARED DIFFERENCES
*
*                  OPEN STUDENT DATA FILE (ENCODED)
*
```

```
      OPEN(11,FILE='STU.DAT',STATUS='OLD')
*
*                  OPEN PRINTER AS FILE
*
      OPEN(6,FILE='PRN',STATUS='NEW')
*
*                  ELIMINATE TRACE (IDBUG) IN FINAL VERSION
*
      IDBUG=0
      CALL SKIP(30)
      WRITE(*,*)'                WELCOME TO THE MAIN SIMULATION',
     +' PROGRAM'
      WRITE(*,*)'                                                '
      WRITE(*,*)'                                 FOR'
      WRITE(*,*)'                                               '
      WRITE(*,*)'                     RESPONSE SURFACE METHODOLOGY'
      CALL SKIP(10)
  1   WRITE(*,*)'                        PLEASE ENTER YOUR NAME '
      CALL SKIP(2)
      READ(*,4,ERR=1) NAME
  3   FORMAT(I1)
  4   FORMAT(A25)
      CALL SKIP(24)
      SEED=12345.D0
*
*                  READ ENCODED VALUES FROM STUDENT DATA FILE
*
      REWIND(11)
      READ(11,*) NSUR
      DO 15, I=1,NSUR
        READ(11,*) (SCOEF(I,J),J=1,5)
        READ(11,*) (SCIACT(I,J),J=1,10)
        READ(11,*) (SCFORD(I,J),J=1,5)
        READ(11,*) SCONST(I)
        READ(11,*) EVAR(I)
        READ(11,*) (START(I,J),J=1,5)
*
*                  DECODE DATA FROM STU.DAT
*
      DO 7 J=1,5
        COEF(I,J)= (SCOEF(I,J)/2)-14
        CFORD(I,J)= (SCFORD(I,J)/3)-21
  7   CONTINUE
      DO 10 J=1,10
        CIACT(I,J)= (SCIACT(I,J)/4)-11
 10   CONTINUE
      CONST(I)= (SCONST(I)/7)-5
 15   CONTINUE
      REWIND(11)
*
*                  BEGIN STUDENT INPUT
*
 20   WRITE(*,21)
 21   FORMAT(10X,'WHAT SURFACE NUMBER HAVE YOU BEEN ASSIGNED [1-15]?')
      CALL SKIP(12)
      READ(*,*,ERR=20) NSTUD
      CALL SKIP(24)
      WRITE(*,*)
      IF (IDBUG.EQ.1) THEN
         WRITE(6,*) (COEF(NSTUD,J),J=1,5)
         WRITE(6,*) (CIACT(NSTUD,J),J=1,10)
         WRITE(6,*) (CFORD(NSTUD,J),J=1,5)
         WRITE(6,*) CONST(NSTUD)
         WRITE(6,*) EVAR(NSTUD)
         WRITE(6,*) (START(NSTUD,J),J=1,5)
         WRITE(*,*)
         CALL CONT
 22   ENDIF
      IF (NSTUD.LE.NSUR) GOTO 25
      WRITE(*,*)
      WRITE(*,*)'           THERE IS NO DATA FOR YOUR SURFACE NUMBER'
      GOTO 20
      WRITE(*,*)
  25  WRITE(*,*)'               HOW MANY INDEPENDENT VARIABLES (1-5)?'
      WRITE(*,*)
      CALL SKIP(12)
      READ(*,*,ERR=25) NVAR
      WRITE(*,*)'  YOUR FIRST DESIGN POINT IS (X1,X2,...):'
      CALL SKIP(3)
      WRITE(*,28) (START(NSTUD,I),I=1,NVAR)
  28  FORMAT(10X,5F12.4)
      CALL SKIP(7)
      CALL CONT
      CALL SKIP(24)
  30  WRITE(*,32)
  32  FORMAT(8X,'HOW MANY "DISTINCT" DESIGN POINTS IN THIS',
     +' EXPERIMENT?')
      CALL SKIP(12)
      READ(*,*,ERR=30) NDOBS
      WRITE(*,*)
      CALL SKIP(23)
  35  WRITE(*,*)'                  WHAT ORDER EQUATION ARE YOU USING?'
      WRITE(*,*)
      WRITE(*,*)'                        "1"= FIRST ORDER'
      WRITE(*,*)'                        "2"= SECOND ORDER'
      CALL SKIP(9)
      READ(*,*,ERR=35) IORDER
      IF(IORDER.NE.1.and.IORDER.NE.2) GOTO 35
      WRITE(*,*)
      CALL SKIP (23)
*
*                  TOTAL VARIABLES= 1+INDEP VAR
      NV=NVAR + 1
*
*                  INPUT DATA
*
  40  WRITE(*,*)'                  BEGIN STUDENT DATA INPUT'
      WRITE(*,*)
      NOBS=0
  45  DO 80 I=1,NDOBS
      WRITE(*,*)
  50  WRITE(*,55) I
  55  FORMAT(9X,'HOW MANY TOTAL REPLICATIONS AT POINT #',I2)
      CALL SKIP(2)
      READ(*,*,ERR=50) NREP
      NREP=NOBS+NREP
       DO 65 J=2,NV
        WRITE(*,60) I,J-1
  60    FORMAT(1X,'POINT # ',I2,'--X',I1,' = ')
        READ(*,*,ERR=50) Z(I,J)
        CALL SKIP(2)
  65   CONTINUE
          DO 75 N=(NOBS+1),NREP
            DO 70 J=2,NV
```

```
              X(N,J)= (Z(I,J))
   70       CONTINUE
              X(N,1)=1.0
              NPT(N)=I
   75     CONTINUE
          NOBS=NREP
*
      CALL SKIP(5)
   80 CONTINUE
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)'                      SUMMARY OF DATA INPUT'
      WRITE(*,*)
      WRITE(*,85) NOBS
   85 FORMAT(17X,'TOTAL OBSERVATIONS= ',I2)
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)'OB#  PT#     DUMMY      X1         X2         X3',
     +'         X4         X5'
      DO 95 I=1,NOBS
        WRITE(*,90) I, NPT(I), (X(I,J),J=1,NV)
   90   FORMAT(1X,I2,2X,I2,2X,6(2X,F8.3))
        IF(I.EQ.19.OR.I.EQ.38.OR.I.EQ.57) THEN
          WRITE(*,*)
          CALL CONT
        ENDIF
   95 CONTINUE
      WRITE(*,*)
  100 WRITE(*,*)'                         IS THIS DATA CORRECT?'
      WRITE(*,*)'                            "1"= CORRECT'
      WRITE(*,*)'                            "0"= REENTER DATA'
      READ(*,*,ERR=100) IANS
      IF (IANS.EQ.1) GOTO 105
      IF ((IANS.NE.1).AND.(IANS.NE.0)) GOTO 100
      WRITE(*,*)'                 PLEASE REENTER YOUR DATA '
      CALL SKIP(15)
      GO TO 20
  105 CONTINUE
*
*                   GENERATE VALUE OF DEPENDENT VARIABLE AT EACH POINT
*
      DO 120 I=1,NOBS
*       DETERMINE DEVIATION
        CALL RANNUM(SEED,RA)
        CALL RANNUM(SEED,RB)
        V=(-2.0*ALOG(RA))**0.5*COS(6.283*RB)
        RNORM=V*(SQRT(EVAR(NSTUD)))
        DEV=RNORM
        IF (RNORM.LT.(-4*SQRT(EVAR(NSTUD)))) THEN
           DEV= -4.0*SQRT(EVAR(NSTUD))
        ENDIF
        IF (RNORM.GT.(4*SQRT(EVAR(NSTUD)))) THEN
           DEV= 4.0*SQRT(EVAR(NSTUD))
        ENDIF
        IF (IDBUG.EQ.1) THEN
*          WRITE(6,*)'RA/RB=',RA,RB
           WRITE(6,*)'V= ',V
*          WRITE(6,*)'EVAR(NSTUD)= ',EVAR(NSTUD)
*          WRITE(6,*)'RNORM= ',RNORM
           WRITE(6,*)'DEV= ',DEV
           WRITE(6,*)
  113     ENDIF
```

```
        Y(I)=0
        DO 115 K=1,NVAR
           Y(I)=Y(I)+COEF(NSTUD,K)*X(I,K+1)**2
     +     +CFORD(NSTUD,K)*X(I,K+1)
 115    CONTINUE
        Y(I)=Y(I)+CIACT(NSTUD,1)*X(I,2)*X(I,3)+
     +  CIACT(NSTUD,2)*X(I,2)*X(I,4)+CIACT(NSTUD,3)*X(I,2)*X(I,5)+
     +  CIACT(NSTUD,4)*X(I,2)*X(I,6)+CIACT(NSTUD,5)*X(I,3)*X(I,4)+
     +  CIACT(NSTUD,6)*X(I,3)*X(I,5)+CIACT(NSTUD,7)*X(I,3)*X(I,6)+
     +  CIACT(NSTUD,8)*X(I,4)*X(I,5)+CIACT(NSTUD,9)*X(I,4)*X(I,6)+
     +  CIACT(NSTUD,10)*X(I,5)*X(I,6)+CONST(NSTUD)+DEV
*
      IF (IDBUG.EQ.1) WRITE(6,*) Y(I)
 120  CONTINUE
      CALL SKIP(25)
*     WRITE(*,130) (I,Y(I),I=1,NOBS)
 130  FORMAT(15X,'Y(',I2,')= ',F18.9)
*
*                  ARRANGE DATA IN ORDER FOR MANIPULATION
*
      NVS=NV
      IF (IORDER.EQ.2) NVS=2*NVAR+1+ICOM(NVAR)
*
*                  ARRANGEMENT UNNECESSARY FOR FIRST ORDER EQUATION
*
      IF(IORDER.EQ.1) GOTO 200
*     NV1=NVAR+1
      DO 170 J=1, NOBS
        DO 160 K=2,NVAR+1
         JJ=K+NVAR
         X(J,JJ)= X(J,K)**2
 160    CONTINUE
 170  CONTINUE
      DO 180 J=1,NOBS
        KK=2*NVAR+2
        X(J,KK)=X(J,2)*X(J,3)
        IF(NVAR.LT.3) GOTO 180
        KK=KK+1
        X(J,KK)=X(J,2)*X(J,4)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,4)
        IF(NVAR.LT. 4) GOTO 180
        X(J,KK)=X(J,2)*X(J,5)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,4)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,5)
        KK=KK+1
        X(J,KK)=X(J,4)*X(J,5)
        IF (NVAR.LT.5) GOTO 180
        KK=KK-2
        X(J,KK)=X(J,2)*X(J,6)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,4)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,5)
        KK=KK+1
        X(J,KK)=X(J,3)*X(J,6)
        KK=KK+1
        X(J,KK)=X(J,4)*X(J,5)
        KK=KK+1
        X(J,KK)=X(J,4)*X(J,6)
```

```
          KK=KK+1
          X(J,KK)=X(J,5)*X(J,6)
  180   CONTINUE
  200   CONTINUE
        WRITE(*,*)'                  -----REARRANGED INPUT DATA-----'
        WRITE(*,*)
        DO 210 I=1,NOBS
          WRITE(*,220) Y(I),NPT(I),(X(I,LL),LL=1,NVS)
          IF(I.EQ.19.OR.I.EQ.38.OR.I.EQ.57) CALL CONT
  210   CONTINUE
  220   FORMAT(1X,F15.8,I2,F7.4,21F11.4)
            WRITE(*,*)
            WRITE(*,*)
            WRITE(*,*)
            CALL CONT
*
*                    BEGIN ANOVA COMPUTATIONS
*
*                    TAKE TRANSPOSE OF MATRIX X, CALL IT MATRIX XT
*
  225   NCOL=NOBS
        NROW=NVS
        DO 250 J=1,NCOL
          DO 240 I=1,NROW
           XT(I,J)=X(J,I)
  240     CONTINUE
  250   CONTINUE
*
        IF (IDBUG.EQ.1) THEN
          WRITE(6,*)'    ------MATRIX XT-----'
          DO 254 I=1,NVS
           WRITE(6,258) (XT(I,LL),LL=1,NOBS)
  254     CONTINUE
  258     FORMAT(1X,50F11.3)
        ENDIF
*
*                    MULTIPLY MATRIX XT * MATRIX X= MATRIX A
*
  259   IXTCOL=NOBS
        IAROW=NVS
        IACOL=NVS
*
        DO 280 I=1,IAROW
          DO 270 J=1,IACOL
            A(I,J)=0.0
            DO 260 K=1,IXTCOL
              A(I,J)= A(I,J) + XT(I,K)*X(K,J)
  260       CONTINUE
  270     CONTINUE
  280   CONTINUE
*
        IF(IDBUG.EQ.1)THEN
          WRITE(6,*)'    -----MATRIX XT*X = MATRIX A-----'
          DO 290 I=1,NVS
            WRITE(6,300) (A(I,J),J=1,NVS)
  290     CONTINUE
  300     FORMAT(1X,50F13.5)
        ENDIF
*
*                    MULTIPLY MATRIX XT * MATRIX Y= MATRIX G
*
  305   IXTCOL=NOBS
```

```
      IGROW=NVS
*
      DO 320 I=1,IGROW
        G(I)=0.0
        DO 310 K=1, IXTCOL
          G(I)=G(I) + XT(I,K)*Y(K)
 310    CONTINUE
 320  CONTINUE
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'   -----MATRIX XT*Y = MATRIX G-----'
        DO 330 I=1,NVS
          WRITE(6,340) G(I)
 330    CONTINUE
 340    FORMAT(1X,F13.5)
      ENDIF
*
*                  TAKE INVERSE OF MATRIX A = XT*X
*
 345  DO 360 I=1, NVS
        DO 350 J=1, NVS
          AA(I,J)=A(I,J)
 350    CONTINUE
 360  CONTINUE
*
*                  SET SIZE OF AINV(I,J) TO THAT OF AA(I,J)
*
      DO 380 I=1, NVS
        DO 370 J=1, NVS
          IF (I.EQ.J) THEN
            AINV(I,J) =1.0
          ELSE
            AINV(I,J)= 0.0
          ENDIF
 370    CONTINUE
 380  CONTINUE
*
*                  INVERT MATRIX USING ROW OPERATIONS
*
      DO 420 I=1,NVS
        CON= AA(I,I)
        IF (CON.EQ.0) THEN
          WRITE(*,*)'              ZERO ON DIAGONAL'
          WRITE(*,*)'              RECHECK EXPERIMENTAL DESIGN'
          WRITE(*,*)
          STOP
        ENDIF
        DO 390 J=1,NVS
          AA(I,J)= AA(I,J)/CON
          AINV(I,J)=AINV(I,J)/CON
 390    CONTINUE
        DO 410 K=1,NVS
          VAL= AA(K,I)
          IF (K.NE.I) THEN
            DO 400 J=1,NVS
              AA(K,J)= (AA(K,J) - AA(I,J)*VAL)
              AINV(K,J)=(AINV(K,J)-AINV(I,J)*VAL)
 400        CONTINUE
          ENDIF
 410    CONTINUE
 420  CONTINUE
*
```

```
*                  PRINT INVERSE OF MATRIX A= AINV
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'    -----MATRIX AINV = INVERSE OF MATRIX A-----'
        DO 430 I=1,NVS
          WRITE(6,440) (AINV(I,J),J=1,NVS)
 430    CONTINUE
 440    FORMAT(1X,25F20.5)
      ENDIF
*
*                  SEE THAT MATRIX AINV*A = DIAGONAL MATRIX
*
 445  DO 47C I=1,NVS
        DO 460 J=1,NVS
          TEST(I,J)=0.0
          DO 450 K=1,NVS
            TEST(I,J)= TEST(I,J) + A(I,K)*AINV(K,J)
*           IF (TEST(I,J).LT.1.0E-06.AND.TEST(I,J).GT.-1.0E-06)
*    +      TEST(I,J)=0.0
 450      CONTINUE
 460    CONTINUE
 470  CONTINUE
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'    -----MATRIX A*AINV = MATRIX TEST-----'
        DO 480 I=1,NVS
          WRITE(6,490) (TEST(I,J),J=1,NVS)
 480    CONTINUE
 490    FORMAT(1X,50F22.5)
      ENDIF
*
*                 MULTIPLY MATRIX AINV*G= MATRIX B(COEFFICIENT MATRIX)
*
 495  DO 520 I=1,NVS
        B(I)=0.0
        DO 500 K=1,NVS
          B(I)= B(I) + AINV(I,K)*G(K)
 500    CONTINUE
 520  CONTINUE
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'  -----MATRIX AINV*G = MATRIX B(COEFFICIENT)-----'
        DO 530 I=0,NVS-1
          WRITE(6,540) I,B(I+1)
 530    CONTINUE
 540    FORMAT(11X,'B',I2,'=',F16.5)
      ENDIF
*
*                  SS(1)=TOT SS  IDEF(1)= DF TOT
*                  SS(2)=ßo SS
*                  SET #VARIABLES BACK TO NV
*
 545   NV=NVS
      TEESS=0.0D0
      IDEF(NV+4)=0
      COUNT=0.0
      SS(1)=0.0D0
      DO 600 I=1,NOBS
        SS(1)=Y(I)**2 + SS(1)
 600  CONTINUE
      IDEF(1)=NOBS
      IDEF(2)=1
      SS(2)= (G(1)**2)/NOBS
      DO 610 I=2,NV
        IF (IDBUG.EQ.1) WRITE(6,*) 'A(',I,',',I,')=',A(I,I)
        SS(I+1)= (B(I)**2)/AINV(I,I)
        IDEF(I+1)= 1
 610  CONTINUE
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'SS(1)=',SS(1)
        WRITE(6,*)'DF(1)=',IDEF(1)
        WRITE(6,*)'SS(2)=',SS(2)
        WRITE(6,*)'DF(2)=',IDEF(2)
        DO 615 I=2,NV
          WRITE(6,*) 'SS',I+1,'= ',SS(I+1)
          WRITE(6,*) 'DF',I+1,'= ',IDEF(I+1)
 615    CONTINUE
      ENDIF
      TT= 0.0D0
      KK=NV+1
      K=NV+2
      IF (IORDER.EQ.2) GOTO 640
*
*                   SUM SS FOR ALL VARIABLES
*
      DO 630 I=2,NV+1
        TT=TT + SS(I)
 630  CONTINUE
*
*                   CALC SS RESIDUAL
*
      SS(K)= SS(1) - TT
      GO TO 680
 640  SSR= 0.0D0
      DO 670 I=2,NV
        SUM= 0.0D0
        DO 650 J=1,NOBS
          SUM=SUM +X(J,I)
 650    CONTINUE
        SUM=SUM/NOBS
        SAM=0.0D0
        DO 660 J=1,NOBS
          SAM=SAM +(X(J,I)-SUM)*Y(J)
 660    CONTINUE
        SSR=SSR+SAM*B(I)
 670  CONTINUE
      SSR= SS(1)-SS(2)-SSR
      SS(NV+2)=SSR
      EESS=0.0D0
      ESS=0.0D0
 680  CONTINUE
      DO 700 I=1,NDOBS
        DO 690 J=1,NOBS
          IF (NPT(J).EQ.I) THEN
            EESS= Y(J)**2 +EESS
            ESS=  Y(J) + ESS
            COUNT=COUNT+1
          ENDIF
 690    CONTINUE
            TEESS=(EESS-ESS**2/COUNT) + TEESS
            IDEF(NV+4)=IDEF(NV+4)+ (COUNT-1)
          COUNT=0
          EESS=0.0D0
          ESS=0.0D0
 700  CONTINUE
*
*                 SS(NV+4)- SS ERROR
*
      SS(NV+4)= TEESS
*
*                 SS(NV+3)- SS LOF
*
      SS(NV+3)= SS(NV+2)-SS(NV+4)
      IDEF(NV+2)= IDEF(1)-NV
      IDEF(NV+3)= IDEF(NV+2)-IDEF(NV+4)
      KN=NV+4
      IJLM=0
      IF(IDEF(NV+2).LT.1.OR.IDEF(NV+3).LT.1.OR.IDEF(NV+4).LT.1)IJLM=1
      DO 710 I=1,KN
        TMS(I)=SS(I)/IDEF(I)
 710  CONTINUE
      ISKIP=0
      IF(TMS(K).NE.0) GOTO 720
      ISKIP=1
      TMS(NV+2)=SSR
 720  CONTINUE
      DO 730 I=1,KN
        FRATIO(I)=TMS(I)/TMS(KN)
 730  CONTINUE
      CALL SKIP(25)
      WRITE(*,740)
 740  FORMAT(1X,20X,'ANALYSIS OF VARIANCE TABLE',/)
      WRITE(*,750)
 750  FORMAT(3X,'SOURCE',7X,'DF',8X,'SS',13X,'MS',11X,'F-RATIO',6X,
     +'COEFFICIENT',/)
*
      IF (IORDER.EQ.2) GOTO 820
      WRITE(*,760) IDEF(1),SS(1),TMS(1)
 760  FORMAT(3X,'TOTAL  ',5X,I3,1X,E14.7,1X,E14.7)
      DO 770 I=2,NV+1
        J=I-1
        IF(I.EQ.2) WRITE(*,772) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.3) WRITE(*,773) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.4) WRITE(*,774) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.5) WRITE(*,775) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.6) WRITE(*,776) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.7) WRITE(*,777) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
 770  CONTINUE
 772  FORMAT(3X,'DUE TO ßo',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 773  FORMAT(3X,'DUE TO ß1',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 774  FORMAT(3X,'DUE TO ß2',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 775  FORMAT(3X,'DUE TO ß3',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 776  FORMAT(3X,'DUE TO ß4',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 777  FORMAT(3X,'DUE TO ß5',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
*
      K=NV+2
      WRITE(*,790) IDEF(K),SS(K),TMS(K)
 790  FORMAT(3X,'RESIDUAL ',3X,I3,1X,E14.7,1X,E14.7)
      IF(ISKIP.EQ.1) GOTO 935
      K=NV+3
      WRITE(*,800) IDEF(K),SS(K),TMS(K),FRATIO(K)
 800  FORMAT(4X,'LACK OF FIT',1X,I3,0X,E14.7,1X,E14.7,1X,E14.7)
      K=NV+4
      WRITE(*,810) IDEF(K),SS(K),TMS(K)
 810  FORMAT(4X,'ERROR      ',1X,I3,0X,E14.7,1X,E14.7)
      GO TO 945
 820  K=1
      WRITE(*,900) IDEF(K),SS(K),TMS(K)
      K=K+1
      WRITE(*,901) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      K=K+1
      WRITE(*,902) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.1) GOTO 840
      K=K+1
      WRITE(*,903) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 840
      K=K+1
      WRITE(*,904) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 840
      K=K+1
      WRITE(*,905) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 840
      K=K+1
      WRITE(*,906) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 840  K=K+1
      WRITE(*,907) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.1) GOTO 880
      K=K+1
      WRITE(*,908) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 850
      K=K+1
      WRITE(*,909) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 850
      K=K+1
      WRITE(*,910) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 850
      K=K+1
      WRITE(*,911) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 850  K=K+1
      WRITE(*,912) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 880
      K=K+1
      WRITE(*,913) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 860
      K=K+1
      WRITE(*,914) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 860
      K=K+1
      WRITE(*,915) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 860  K=K+1
      WRITE(*,916) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 880
      K=K+1
      WRITE(*,917) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 870
      K=K+1
      WRITE(*,918) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 870  K=K+1
      WRITE(*,919) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 880
      K=K+1
      WRITE(*,920) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      K=K+1
      WRITE(*,921) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 880  K=K+1
      WRITE(*,922) IDEF(K),SS(K),TMS(K)
      K=K+1
      IF(ISKIP.EQ.1) GOTO 935
      WRITE(*,923) IDEF(K),SS(K),TMS(K),FRATIO(K)
      K=K+1
      WRITE(*,924) IDEF(K),SS(K),TMS(K)
*
 900  FORMAT(3X,'TOTAL  ',5X,I3,1X,E14.7,1X,E14.7)
 901  FORMAT(3X,'DUE TO ßo',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 902  FORMAT(3X,'DUE TO ß1',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 903  FORMAT(3X,'DUE TO ß2',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 904  FORMAT(3X,'DUE TO ß3',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 905  FORMAT(3X,'DUE TO ß4',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 906  FORMAT(3X,'DUE TO ß5',3X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 907  FORMAT(3X,'DUE TO ß11',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 908  FORMAT(3X,'DUE TO ß22',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 909  FORMAT(3X,'DUE TO ß33',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 910  FORMAT(3X,'DUE TO ß44',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 911  FORMAT(3X,'DUE TO ß55',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 912  FORMAT(3X,'DUE TO ß12',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 913  FORMAT(3X,'DUE TO ß13',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 914  FORMAT(3X,'DUE TO ß14',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 915  FORMAT(3X,'DUE TO ß15',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 916  FORMAT(3X,'DUE TO ß23',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 917  FORMAT(3X,'DUE TO ß24',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 918  FORMAT(3X,'DUE TO ß25',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 919  FORMAT(3X,'DUE TO ß34',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 920  FORMAT(3X,'DUE TO ß35',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 921  FORMAT(3X,'DUE TO ß45',2X,I3,1X,E14.7,1X,E14.7,1X,E14.7,1X,
     +E14.7)
 922  FORMAT(3X,'RESIDUAL ',3X,I3,1X,E14.7,1X,E14.7)
 923  FORMAT(4X,'LACK OF FIT',1X,I3,0X,E14.7,1X,E14.7,1X,E14.7)
 924  FORMAT(4X,'ERROR      ',1X,I3,0X,E14.7,1X,E14.7,//)
      IF(IJLM.EQ.1) WRITE(*,930)
 930  FORMAT(//,10X,'NOT ENOUGH POINTS TO ESTIMATE ALL PARAMETERS')
      GOTO 945
 935  WRITE(*,940)
 940  FORMAT(//,10X,'**EXPERIMENT INSUFFICIENT TO ESTIMATE LACK OF FIT
     +AND EXPERIMENTAL ERROR')
 945  CONTINUE
      WRITE(*,*)
      CALL CONT
      YDTOTAL=0.0D0
 947  DO 960 K=1,NOBS
        YF(K)=0.0D0
        DO 950 I=1,NVS
          YF(K)=YF(K)+B(I)*X(K,I)
 950    CONTINUE
        YD(K)=Y(K)-YF(K)
        YDSQ(K)=YD(K)**2
        YDTOTAL=YDTOTAL + YDSQ(K)
 960  CONTINUE
      CALL SKIP(5)
      WRITE(*,970)
 970  FORMAT(5X,'POINT',3X,'GENERATED',7X,'FORECASTED',6X,
     +'DIFFERENCE',6X,'DIFF SQUARED')
      DO 975 K=1,NOBS
        WRITE(*,980) NPT(K),Y(K),YF(K),YD(K),YDSQ(K)
        IF(K.EQ.19.OR.K.EQ.38.OR.K.EQ.57) CALL CONT
 975  CONTINUE
 980  FORMAT(5X,I2,3X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7)
*
      WRITE(*,981) YDTOTAL
 981  FORMAT(30X,'SUM OF SQUARED DIFFERENCES= ',E14.7)
      CALL SKIP(2)
 985  WRITE(*,*)'   DO YOU WANT PRINTED COPY OF RESULTS?'
      WRITE(*,*)
      WRITE(*,*)'   YOU MUST PUT PRINTER ON LINE FOR PRINTED RESULTS'
      CALL SKIP(3)
      WRITE(*,*)'           HARD COPY=1   SCREEN ONLY=0'
      READ(*,3,ERR=985) ILIST
      CALL SKIP(24)
*
      IF(ILIST.EQ.1) THEN
        WRITE(6,*) '  NAME:',NAME
        WRITE(6,*)
        WRITE(6,990) NSTUD
 990    FORMAT(3X,'SURFACE NUMBER: ',I2)
        WRITE(6,*)
        WRITE(6,*)'             -----REARRANGED INPUT DATA-----'
        WRITE(6,*)
        DO 995 I=1,NOBS
          WRITE(6,1000) Y(I),NPT(I),(X(I,LL),LL=1,NV)
 995    CONTINUE
 1000   FORMAT(1X,F15.8,I2,F7.4,21F11.4)
      ENDIF
*
*                  PRINT COPY OF MATRIX B(COEFFICIENTS)
*
      IF(ILIST.EQ.1) THEN
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,740)
      WRITE(6,750)
      IF (IORDER.EQ.2) GOTO 1820
      WRITE(6,760) IDEF(1),SS(1),TMS(1)
      DO 1770 I=2,NV+1
        J=I-1
        IF(I.EQ.2) WRITE(6,772) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.3) WRITE(6,773) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.4) WRITE(6,774) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.5) WRITE(6,775) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.6) WRITE(6,776) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
        IF(I.EQ.7) WRITE(6,777) IDEF(I),SS(I),TMS(I),FRATIO(I),B(J)
 1770  CONTINUE
*
      K=NV+2
      WRITE(6,790) IDEF(K),SS(K),TMS(K)
      IF(ISKIP.EQ.1) GOTO 1935
      K=NV+3
      WRITE(6,800) IDEF(K),SS(K),TMS(K),FRATIO(K)
      K=NV+4
      WRITE(6,810) IDEF(K),SS(K),TMS(K)
      GO TO 1945
 1820 K=1
      WRITE(6,900) IDEF(K),SS(K),TMS(K)
      K=K+1
      WRITE(6,901) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      K=K+1
      WRITE(6,902) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.1) GOTO 1840
      K=K+1
      WRITE(6,903) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 1840
      K=K+1
      WRITE(6,904) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 1840
      K=K+1
      WRITE(6,905) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 1840
      K=K+1
      WRITE(6,906) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 1840 K=K+1
      WRITE(6,907) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.1) GOTO 1880
      K=K+1
      WRITE(6,908) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 1850
      K=K+1
      WRITE(6,909) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 1850
      K=K+1
      WRITE(6,910) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 1850
      K=K+1
      WRITE(6,911) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 1850 K=K+1
      WRITE(6,912) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.2) GOTO 1880
      K=K+1
      WRITE(6,913) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 1860
      K=K+1
      WRITE(6,914) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 1860
      K=K+1
      WRITE(6,915) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 1860 K=K+1
      WRITE(6,916) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.3) GOTO 1880
      K=K+1
      WRITE(6,917) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 1870
      K=K+1
      WRITE(6,918) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 1870 K=K+1
      WRITE(6,919) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      IF(NVAR.EQ.4) GOTO 1880
      K=K+1
      WRITE(6,920) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
      K=K+1
      WRITE(6,921) IDEF(K),SS(K),TMS(K),FRATIO(K),B(K-1)
 1880 K=K+1
      WRITE(6,922) IDEF(K),SS(K),TMS(K)
      K=K+1
      IF(ISKIP.EQ.1) GOTO 1935
      WRITE(6,923) IDEF(K),SS(K),TMS(K),FRATIO(K)
      K=K+1
      WRITE(6,924) IDEF(K),SS(K),TMS(K)
*
      IF(IJLM.EQ.1) WRITE(*,930)
      GOTO 1945
1935  WRITE(6,940)
1945  CONTINUE
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,1970)
1970  FORMAT(5X,'POINT',3X,'GENERATED',7X,'FORECASTED',6X,'DIFFERENCE'
     +,6X,'DIFF SQUARED')
      WRITE(6,*)
      DO 1975 K=1,NOBS
        WRITE(6,1980) NPT(K),Y(K),YF(K),YD(K),YDSQ(K)
1975  CONTINUE
1980  FORMAT(5X,I2,3X,E14.7,2X,E14.7,2X,E14.7,2X,E14.7)
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,1985) YDTOTAL
1985  FORMAT(30X,'SUM OF SQUARED DIFFERENCES= ',E14.7)
      WRITE(6,*)
       ENDIF
*
      WRITE(*,*)
      WRITE(*,*)
1990  WRITE(*,*)'          DO YOU WANT TO PERFORM ANOTHER ITERATION?'
      WRITE(*,*)
      WRITE(*,*)'                            "1"= CONTINUE'
      WRITE(*,*)'                            "0"= STOP'
      CALL SKIP(9)
      READ(*,*,ERR=1990) IANS
      IF (IANS.EQ.1) GOTO 30
      IF (IANS.NE.0) GOTO 1990
       STOP
       END
*
*                  RANDOM NUMBER SUBROUTINE
*
      SUBROUTINE RANNUM(SEED,RRAN)
      REAL*8 PROD,SEMI,SEED
      PROD=(16807.D0*SEED)
      SEMI=DMOD(PROD,2147483647.D0)
      RRAN=SEMI * .4656613E-9
      SEED=SEMI
      RETURN
      END
*
*                  SUBROUTINE SKIP-- PRINTS 'N' BLANK LINES
*
      SUBROUTINE SKIP(N)
      DO 10 I=1,N
         WRITE(*,*)' '
   10 CONTINUE
      RETURN
      END
*
```

```
*                  SUBROUTINE CONT- HALT EXECUTION UNTIL USER READY
*
      SUBROUTINE CONT
      CHARACTER*1 ANS,BLK
      DATA BLK/' '/
      ANS=BLK
      WRITE(*,1)
    1 FORMAT(/,'                    To continue, press RETURN key')
      READ(*,2) ANS
    2 FORMAT(A1)
      RETURN
      END
*
      FUNCTION ICOM(NVAR)
      GO TO (1,2,3,4,5),NVAR
 1    ICOM=0
      RETURN
 2    ICOM=1
      RETURN
 3    ICOM=3
      RETURN
 4    ICOM=6
      RETURN
 5    ICOM=10
      RETURN
      END
```

APPENDIX C- CRIT.EXE PROGRAM LISTING

```
*              -------PROGRAM- CRIT.FOR-------
*                                    WRITTEN BY JIM TREHARNE
*                                    MARCH 11, 1991
*
*     THIS PROGRAM IS USED FOR THREE PURPOSES:
*       1- CALCULATE CRITICAL VALUES OF THE SECOND ORDER EQUATION
*       2- ESTIMATE THE MAX/MIN VALUE OF THE DEPENDENT VARIABLE
*       3- PROVIDE DATA TO MAP RESPONSE CONTOURS
*
      PROGRAM CRIT
      DOUBLE PRECISION A(5,5),R(5),COEF(5),CIACT(10),CFORD(5),CONST,
     +                 X(5),AINV(5,5),TEST(5,5),Y,CONTOUR,RITE(5),
     +                 CHECK,Z(5),T(5),P,PO,D,VMIN(5),VMAX(5),DELTA(5)
      INTEGER NSTUD,NVAR,ILIST
      CHARACTER NAME*25
*
*                DEFINITION OF VARIABLES
*
*       A(5,5)- MATRIX USED TO SOLVE FOR INDEPENDENT VARIABLES
*       R(5)- USED IN MATRIX INVERSION
*       COEF(5)- COEFFICIENTS OF HIGHER ORDER TERMS
*       CIACT(10)- COEFFICIENTS OF THE INTERACTION TERMS
*       CFORD(5)- COEFFICIENTS OF FIRST ORDER TERMS
*       CONST(5)- CONSTANT TERM
*       X(5)- OPTIMAL VALUES OF THE INDEPENDENT VARIABLES
*       AINV(5)- INVERSE MATRIX TO SOLVE FOR OPTIMAL VALUES
*       TEST(5,5)- USED TO VERIFY INVERSE MATRIX IN DEBUGGING
*       Y- RESPONSE VARIABLE
*       CONTOUR- VALUE AT WHICH DATA FOR CONTOUR IS DESIRED
*       RITE(5)- RIGHT SIDE COEFFICIENTS
*       VMIN(5)- MINIMUM VALUE OF VARIABLE USED TO MAP CONTOUR
*       VMAX(5)- MAXIMUM VALUE OF VARIABLE USED TO MAP CONTOUR
*       DELTA(5)- USED IN GETTING DATA TO MAP CONTOUR
*       CHECK- USED TO PLOT CONTOUR DATA
*       Z(5)- USED TO PLOT CONTOUR DATA
*       T(5)- USED TO INVERT MATRIX
*       P- PIVOT VALUE USED IN MATRIX INVERSION
*       PO- 1/P
*       D- VALUE OF DETERMINANT USED IN DEBUGGING
*       NSTUD- STUDENT/SURFACE NUMBER
*       NVAR- NUMBER OF INDEPENDENT VARIABLES
*       ILIST- USED TO DETERMINE IF PRINTOUT DESIRED
*
*                OPEN PRINTER AS FILE #6
*
      OPEN(6,FILE='PRN',STATUS='NEW')
*
*                SCREEN START UP INFORMATION
*
      CALL SKIP(10)
      WRITE(*,10)
   10 FORMAT(17X,'WELCOME TO THE CRIT PROGRAM')
```

```
      CALL SKIP(2)
      WRITE(*,*)'        This program is used in conjunction with the'
      WRITE(*,*)
      WRITE(*,*)'        main program entitled RSM.FOR.  This program'
      WRITE(*,*)
      WRITE(*,*)'        will perform three functions:'
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)'          1- Calculate Critical Values of the Second'
      WRITE(*,*)'             Order Equation'
      WRITE(*,*)
      WRITE(*,*)'          2- Estimate the Max/Min Value of the '
      WRITE(*,*)'             Dependent Variable'
      WRITE(*,*)
      WRITE(*,*)'          3- Provide Data to Map Response Contours'
      WRITE(*,*)
      WRITE(*,*)
      CALL SKIP(1)
      CALL CONT
   20 CALL SKIP(23)
   30 WRITE(*,*)'                     PLEASE ENTER YOUR NAME'
      CALL SKIP(12)
      READ(*,40,ERR=30) NAME
   40 FORMAT(A25)
      CALL SKIP(13)
*
*                  INITIALIZE VARIABLES
*
*                  IDBUG=0.0 FOR FINAL PROGRAM
*
   50 IDBUG=0.0
      NSTUD=0
      Y=0.0D0
      NVAR=0
      ILIST=0
      CONST=0
      DO 70 I=1,5
        DO 60 J=1,5
          A(I,J)=0
   60   CONTINUE
        R(I)=0
        COEF(I)=0
        CFORD(I)=0
   70 CONTINUE
      DO 80 I=1,10
        CIACT(I)=0
   80 CONTINUE
   90 WRITE(*,*)'   WHAT SURFACE NUMBER ARE YOU WORKING WITH [1-15] ?'
      CALL SKIP(12)
      READ(*,100,ERR=90) NSTUD
  100 FORMAT(I2)
  110 WRITE(*,*)'              HOW MANY INDEPENDENT VARIABLES [1-5] ?'
```

```
      CALL SKIP(12)
      READ(*,120,ERR=110) NVAR
  120 FORMAT(I2)
      CALL SKIP(24)
      DO 150 I=1,NVAR
  130   WRITE(*,140) I,I
  140   FORMAT(19X,'WHAT IS THE VALUE OF ß',I1,I1,'?')
        CALL SKIP(3)
        READ(*,*,ERR=130) COEF(I)
  150 CONTINUE
      CALL SKIP(24)
*
      IF(NVAR.EQ.2) THEN
  160   WRITE(*,*)'                            WHAT IS THE VALUE OF ß12'
        CALL SKIP(3)
        READ(*,*,ERR=160) CIACT(1)
      ENDIF
      CALL SKIP(24)
*
      IF(NVAR.EQ.3) THEN
  170   WRITE(*,*)'                            WHAT IS THE VALUE OF ß12'
        CALL SKIP(3)
        READ(*,*,ERR=170) CIACT(1)
        CALL SKIP(3)
  180   WRITE(*,*)'                            WHAT IS THE VALUE OF ß13'
        CALL SKIP(3)
        READ(*,*,ERR=180) CIACT(2)
  190   WRITE(*,*)'                            WHAT IS THE VALUE OF ß23'
        CALL SKIP(3)
        READ(*,*,ERR=190) CIACT(5)
        CALL SKIP(24)
      ENDIF
*
      IF(NVAR.EQ.4) THEN
  200   WRITE(*,*)'                            WHAT IS THE VALUE OF ß12'
        CALL SKIP(3)
        READ(*,*,ERR=200) CIACT(1)
        CALL SKIP(3)
  210   WRITE(*,*)'                            WHAT IS THE VALUE OF ß13'
        CALL SKIP(3)
        READ(*,*,ERR=210) CIACT(2)
  220   WRITE(*,*)'                            WHAT IS THE VALUE OF ß14'
        CALL SKIP(3)
        READ(*,*,ERR=220) CIACT(3)
  230   WRITE(*,*)'                            WHAT IS THE VALUE OF ß23'
        CALL SKIP(3)
        READ(*,*,ERR=230) CIACT(5)
  240   WRITE(*,*)'                            WHAT IS THE VALUE OF ß24'
        CALL SKIP(3)
        READ(*,*,ERR=240) CIACT(6)
  250   WRITE(*,*)'                            WHAT IS THE VALUE OF ß34'
        CALL SKIP(3)
        READ(*,*,ERR=250) CIACT(8)
        CALL SKIP(24)
      ENDIF
*
      IF(NVAR.EQ.5) THEN
 260    WRITE(*,*)'                  WHAT IS THE VALUE OF ß12'
        CALL SKIP(3)
        READ(*,*,ERR=260) CIACT(1)
        CALL SKIP(3)
 270    WRITE(*,*)'                  WHAT IS THE VALUE OF ß13'
        CALL SKIP(3)
        READ(*,*,ERR=270) CIACT(2)
 280    WRITE(*,*)'                  WHAT IS THE VALUE OF ß14'
        CALL SKIP(3)
        READ(*,*,ERR=280) CIACT(3)
 290    WRITE(*,*)'                  WHAT IS THE VALUE OF ß15'
        CALL SKIP(3)
        READ(*,*,ERR=290) CIACT(4)
 300    WRITE(*,*)'                  WHAT IS THE VALUE OF ß23'
        CALL SKIP(3)
        READ(*,*,ERR=300) CIACT(5)
 310    WRITE(*,*)'                  WHAT IS THE VALUE OF ß24'
        CALL SKIP(3)
        READ(*,*,ERR=310) CIACT(6)
 320    WRITE(*,*)'                  WHAT IS THE VALUE OF ß25'
        CALL SKIP(3)
        READ(*,*,ERR=320) CIACT(7)
 330    WRITE(*,*)'                  WHAT IS THE VALUE OF ß34'
        CALL SKIP(3)
        READ(*,*,ERR=330) CIACT(8)
 340    WRITE(*,*)'                  WHAT IS THE VALUE OF ß35'
        CALL SKIP(3)
        READ(*,*,ERR=340) CIACT(9)
 350    WRITE(*,*)'                  WHAT IS THE VALUE OF ß45'
        CALL SKIP(3)
        READ(*,*,ERR=350) CIACT(10)
        CALL SKIP(24)
      ENDIF
*
      DO 380 I=1,NVAR
 360    WRITE(*,370) I
 370    FORMAT(19X,'WHAT IS THE VALUE OF ß',I1' ?')
        CALL SKIP(3)
        READ(*,*,ERR=360) CFORD(I)
 380  CONTINUE
      CALL SKIP(24)
*
 390  WRITE(*,400)
 400  FORMAT(19X,'WHAT IS THE VALUE OF THE CONSTANT, ßo ?')
      CALL SKIP(3)
      READ(*,*,ERR=390) CONST
      CALL SKIP(24)
```

```
        READ(*,*,ERR=250) CIACT(8)
        CALL SKIP(24)
      ENDIF
*
      IF(NVAR.EQ.5) THEN
 260    WRITE(*,*)'                   WHAT IS THE VALUE OF ß12'
        CALL SKIP(3)
        READ(*,*,ERR=260) CIACT(1)
        CALL SKIP(3)
 270    WRITE(*,*)'                   WHAT IS THE VALUE OF ß13'
        CALL SKIP(3)
        READ(*,*,ERR=270) CIACT(2)
 280    WRITE(*,*)'                   WHAT IS THE VALUE OF ß14'
        CALL SKIP(3)
        READ(*,*,ERR=280) CIACT(3)
 290    WRITE(*,*)'                   WHAT IS THE VALUE OF ß15'
        CALL SKIP(3)
        READ(*,*,ERR=290) CIACT(4)
 300    WRITE(*,*)'                   WHAT IS THE VALUE OF ß23'
        CALL SKIP(3)
        READ(*,*,ERR=300) CIACT(5)
 310    WRITE(*,*)'                   WHAT IS THE VALUE OF ß24'
        CALL SKIP(3)
        READ(*,*,ERR=310) CIACT(6)
 320    WRITE(*,*)'                   WHAT IS THE VALUE OF ß25'
        CALL SKIP(3)
        READ(*,*,ERR=320) CIACT(7)
 330    WRITE(*,*)'                   WHAT IS THE VALUE OF ß34'
        CALL SKIP(3)
        READ(*,*,ERR=330) CIACT(8)
 340    WRITE(*,*)'                   WHAT IS THE VALUE OF ß35'
        CALL SKIP(3)
        READ(*,*,ERR=340) CIACT(9)
 350    WRITE(*,*)'                   WHAT IS THE VALUE OF ß45'
        CALL SKIP(3)
        READ(*,*,ERR=350) CIACT(10)
        CALL SKIP(24)
      ENDIF
*
      DO 380 I=1,NVAR
 360    WRITE(*,370) I
 370    FORMAT(19X,'WHAT IS THE VALUE OF ß',I1' ?')
        CALL SKIP(3)
        READ(*,*,ERR=360) CFORD(I)
 380  CONTINUE
      CALL SKIP(24)
*
 390  WRITE(*,400)
 400  FORMAT(19X,'WHAT IS THE VALUE OF THE CONSTANT, ßo ?')
      CALL SKIP(3)
      READ(*,*,ERR=390) CONST
      CALL SKIP(24)
```

```
*
*                  WRITE EQUATION TO SCREEN FOR VERIFICATION
*
      WRITE(*,410)
 410  FORMAT(//,27X,'SURFACE EQUATION IS:',//)
      IF(NVAR.EQ.1) THEN
        WRITE(*,420) COEF(1),CFORD(1),CONST
 420    FORMAT(3X,'Y= ',F9.3,'X1**2 + ',F9.3,'X1 + ',F9.3,)
      ENDIF
*
      IF(NVAR.EQ.2) THEN
        WRITE(*,430) COEF(1),COEF(2),CIACT(1),CFORD(1),CFORD(2),CONST
 430    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X1*X2 +',F10.3,'X1 ',//,4X,'+ ',F10.3,'X2    + ',F10.3,)
      ENDIF
*
      IF(NVAR.EQ.3) THEN
        WRITE(*,440) COEF(1),COEF(2),COEF(3),CIACT(1),CIACT(2),
     +   CIACT(5),CFORD(1),CFORD(2),CFORD(3),CONST
 440    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X1*X2',//,4X,'+ ',F10.3,'X1*X3 + ',F10.3,
     +  'X2*X3 + ',F10.3,'X1    +',F10.3,'X2',//,4X,'+ ',F10.3,
     +  'X3    + ',F10.3)
      ENDIF
*
      IF(NVAR.EQ.4) THEN
        WRITE(*,450) COEF(1),COEF(2),COEF(3),COEF(4),CIACT(1),
     +  CIACT(2),  CIACT(3),CIACT(5),CIACT(6),CIACT(8),CFORD(1),
     +  CFORD(2),CFORD(3),CFORD(4),CONST
 450    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X4**2',//,4X,'+ ',F10.3,'X1*X2 + ',F10.3,
     +  'X1*X3 + ',F10.3,'X1*X4 +',F10.3,'X2*X3',//,4X,'+ ',F10.3,
     +  'X2*X4 + ',F10.3,'X3*X4 + ',F10.3,'X1    +',F10.3,'X2 ',//,4X,
     +  '+ ',F10.3,'X3    + ',F10.3,'X4    + ',F10.3)
      ENDIF
*
      IF(NVAR.EQ.5) THEN
        WRITE(*,460) COEF(1),COEF(2),COEF(3),COEF(4),COEF(5),CIACT(1),
     +  CIACT(2),CIACT(3),CIACT(4),CIACT(5),CIACT(6),
     +  CIACT(7),CIACT(8),CIACT(9),CIACT(10),CFORD(1),CFORD(2),
     +  CFORD(3),CFORD(4),CFORD(5),CONST
 460    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X4**2',//,4X,'+ ',F10.3,'X5**2 + ',F10.3,
     +  'X1*X2 + ',F10.3,'X1*X3 +',F10.3,'X1*X4',//,4X,'+ ',F10.3,
     +  'X1*X5 + ',F10.3,'X2*X3 + ',F10.3,'X2*X4
     +  +',F10.3,'X2*X5',//,4X,
     +  '+ ',F10.3,'X3*X4 + ',F10.3,'X3*X5 + ',F10.3,'X4*X5 +',F10.3,
     +  'X1',//,4X,'+ ',F10.3,'X2    + ',F10.3,'X3    + ',F10.3,
     +  'X4    +',F10.3,'X5 ',//,4X,'+ ',F10.3)
      ENDIF
      CALL SKIP(3)
 470  WRITE(*,*)'                         IS THE EQUATION CORRECT?'
```

```
      CALL SKIP(2)
      WRITE(*,*)'                                 "1"= CORRECT'
      WRITE(*,*)
      WRITE(*,*)'                                 "0"= WRONG'
      READ(*,*,ERR=470) IANS
      IF (IANS.EQ.0) GO TO 90
      IF (IANS.NE.1) GO TO 470
*
*                  BEGIN SETTING UP MATRICES
*
      CALL SKIP(24)
      DO 480 K=1,NVAR
        RITE(K)=-CFORD(K)
 480  CONTINUE
      DO 490 K=1,NVAR
        A(K,K)= COEF(K)*2
 490  CONTINUE
      IF(NVAR.LT.2) GO TO 500
      A(1,2)= CIACT(1)
      A(2,1)= CIACT(1)
      IF(NVAR.LT.3) GO TO 500
      A(1,3)= CIACT(2)
      A(3,1)= CIACT(2)
      A(2,3)= CIACT(5)
      A(3,2)= CIACT(5)
      IF(NVAR.LT.4) GO TO 500
      A(1,4)= CIACT(3)
      A(4,1)= CIACT(3)
      A(2,4)= CIACT(6)
      A(4,2)= CIACT(6)
      A(3,4)= CIACT(8)
      A(4,3)= CIACT(6)
      IF(NVAR.LT.5) GO TO 500
      A(1,5)= CIACT(4)
      A(5,1)= CIACT(4)
      A(2,5)= CIACT(7)
      A(5,2)= CIACT(7)
      A(3,5)= CIACT(9)
      A(5,3)= CIACT(9)
      A(4,5)= CIACT(10)
      A(5,4)= CIACT(10)
*
 500  CONTINUE
*
*                  TAKE INVERSE OF MATRIX A = AINV
*
*       THE CODE TO TAKE THE INVERSE USES THE GAUSS-JORDAN TECHNIQUE
*       WITH PARTIAL PIVOTING.  IT IS ADAPTED FROM:
*         MICROCOMPUTERS IN NUMERICAL ANALYSIS
*           BY LINDFIELD AND PENNY, 1987
*
*
```

```
      DO 520 I=1,NVAR
         DO 510 J=1,NVAR
            AINV(I,J)=A(I,J)
 510     CONTINUE
 520  CONTINUE
      D=1
      DO 590 K=1,NVAR
*
*                  CHOOSE PIVOTS
*
         P=0
         DO 550 I=1,NVAR
            IF (K.EQ.1) GOTO 540
            DO 530 L=1,(K-1)
               IF (I.EQ.(R(L))) GOTO 550
 530        CONTINUE
 540        IF (DABS(AINV(I,K)).LE.DABS(P)) GOTO 550
            P=AINV(I,K)
            R(K)=I
 550     CONTINUE
         IF (P.EQ.0) THEN
            WRITE(*,*) 'ZERO PIVOT'
            STOP
         ENDIF
         D=D*P
         PO=1/P
*
*                  ELIMINATION PROCEDURE
*
         DO 560 J=1,NVAR
            M=R(K)
            AINV(M,J)=AINV(M,J)*PO
 560     CONTINUE
         AINV(M,K)=PO
         DO 580 I=1,NVAR
            IF (I.EQ.(R(K))) GOTO 580
            DO 570 J=1,NVAR
               IF (J.EQ.K) GOTO 570
               M=R(K)
               AINV(I,J)=AINV(I,J)-AINV(I,K)*AINV(M,J)
 570        CONTINUE
            AINV(I,K)= -AINV(I,K)*PO
 580     CONTINUE
 590  CONTINUE
      DO 620 J=1,NVAR
         DO 600 I=1,NVAR
            M=R(I)
            T(I)=AINV(M,J)
 600     CONTINUE
         DO 610 I=1,NVAR
            AINV(I,J)=T(I)
 610    CONTINUE
 620  CONTINUE
      DO 650 I=1,NVAR
         DO 630 J=1,NVAR
            M=R(J)
            T(M)=AINV(I,J)
 630     CONTINUE
         DO 640 J=1,NVAR
            AINV(I,J)=T(J)
 640     CONTINUE
 650  CONTINUE
      DO 660 K=1,NVAR
         M=R(K)
         T(M)=K
 660  CONTINUE
      DO 680 I=1,NVAR
         DO 670 J=1,(NVAR-1)
            IF (T(J).LE.(T(J+1))) GOTO 670
            P=T(J)
            T(J)=T(J+1)
            T(J+1)=P
            D= -D
 670     CONTINUE
 680  CONTINUE
      IF (IDBUG.EQ.1) THEN
*                  WRITE MATRIX
      DO 690 I=1,NVAR
            WRITE(*,*) (AINV(I,J),J=1,NVAR)
 690  CONTINUE
      WRITE(*,*)
      WRITE(*,*)'DETERMINANT= ',D
*
*                  PRINT INVERSE OF MATRIX A= AINV
*
        WRITE(6,*)'    -----MATRIX AINV = INVERSE OF MATRIX A-----'
        DO 700 I=1,NVAR
          WRITE(6,710) (AINV(I,J),J=1,NVAR)
 700    CONTINUE
 710    FORMAT(1X,5F20.5)
      ENDIF
*
*                  SEE THAT MATRIX AINV*A = DIAGONAL MATRIX
*
 720  DO 750 I=1,NVAR
        DO 740 J=1,NVAR
          TEST(I,J)=0.0
          DO 730 K=1,NVAR
            TEST(I,J)= TEST(I,J) + A(I,K)*AINV(K,J)
*           IF (TEST(I,J).LT.1.0E-06.AND.TEST(I,J).GT.-1.0E-06)
*    +      TEST(I,J)=0.0
 730      CONTINUE
 740    CONTINUE
 750  CONTINUE
```

```
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'    -----MATRIX A*AINV = MATRIX TEST-----'
        DO 760 I=1,NVAR
          WRITE(6,770) (TEST(I,J),J=1,NVAR)
 760    CONTINUE
 770    FORMAT(1X,5F22.5)
      ENDIF
*
*                    MULTIPLY MATRIX AINV*RITE(K)= MATRIX X(VAL MATRIX)
*
 780  DO 800 I=1,NVAR
        X(I)=0.0
        DO 790 K=1,NVAR
          X(I)= X(I) + AINV(I,K)*RITE(K)
 790    CONTINUE
 800  CONTINUE
*
      IF(IDBUG.EQ.1) THEN
        WRITE(6,*)'    -----MATRIX AINV*R(K) = MATRIX X(VALUE)-----'
        DO 810 I=1,NVAR
          WRITE(6,820) I,X(I)
 810    CONTINUE
 820    FORMAT(11X,'X',I2,'=',F16.5)
      ENDIF
*
*                    DISPLAY RESULTS TO SCREEN
*
      WRITE(*,*)'                   CRITICAL ANALYSIS OF SURFACE'
      CALL SKIP(1)

      WRITE(*,*)'--VALUE OF INDEPENDENT VARIABLES AT OPTIMAL POINT--'
      CALL SKIP(1)
      DO 830 I=1,NVAR
        WRITE(*,840) I,X(I)
 830  CONTINUE
 840  FORMAT(11X,'X',I2,'=',F16.6,/)
      CALL SKIP(1)
*
*                    DETERMINE VALUE OF Y AT OPTIMAL POINT
*
      DO 850 K=1,NVAR
        Y=COEF(K)*X(K)**2 + CFORD(K)*X(K) + Y
 850  CONTINUE
      Y= Y+X(1)*X(2)*CIACT(1) + X(1)*X(3)*CIACT(2) +
     +     X(1)*X(4)*CIACT(3) +
     +     X(1)*X(5)*CIACT(4) + X(2)*X(3)*CIACT(5) +
     +     X(2)*X(4)*CIACT(6) +
     +     X(2)*X(5)*CIACT(7) + X(3)*X(4)*CIACT(8) +
     +     X(3)*X(5)*CIACT(9) +
     +     X(4)*X(5)*CIACT(10) + CONST
*
```

```
      WRITE(*,860) Y
 860  FORMAT(2X,'--VALUE OF RESPONSE VARIABLE AT OPTIMAL POINT--',/
     +/,12X,'Y =',F16.6,/)
*
 870  WRITE(*,880)
 880  FORMAT(2X,'--DO YOU WANT PRINTOUT OF THE RESULTS?',/)
      WRITE(*,*)'                      "1"= PRINT '
      WRITE(*,*)'                      "0"= NO'
      WRITE(*,*)
      READ(*,*,ERR=870) IANS
      IF (IANS.NE.0.AND.IANS.NE.1) GO TO 870
      IF (IANS.EQ.0) GO TO 990
*
      CALL SKIP(12)
      WRITE(*,*)'                       PRINTING RESULTS'
      CALL SKIP(12)
      WRITE(6,*)'                 CRITICAL ANALYSIS OF SURFACE'
      WRITE(6,*)
      WRITE(6,*)'NAME: ',NAME
      WRITE(6,*)
      WRITE(6,890) NSTUD
 890  FORMAT(3X,'SURFACE NUMBER= ',I2)
      WRITE(6,900)
 900  FORMAT(/,15X,'SURFACE EQUATION IS:',/)
      IF(NVAR.EQ.1) THEN
        WRITE(6,910) COEF(1),CFORD(1),CONST
 910   FORMAT(3X,'Y= ',F9.3,'X1**2 + ',F9.3,'X1 + ',F9.3,)
      ENDIF
*
      IF(NVAR.EQ.2) THEN
        WRITE(6,920) COEF(1),COEF(2),CIACT(1),CFORD(1),CFORD(2),CONST
 920    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X1*X2 +',F10.3,'X1 ',//,4X,'+ ',F10.3,'X2    + ',F10.3,)
      ENDIF
*
      IF(NVAR.EQ.3) THEN
        WRITE(6,930)COEF(1),COEF(2),COEF(3),CIACT(1),CIACT(2),
     +  CIACT(5),CFORD(1),CFORD(2),CFORD(3),CONST
 930     FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X1*X2',//,4X,'+ ',F10.3,'X1*X3 + ',F10.3,
     +  'X2*X3 + ',F10.3,'X1    +',F10.3,'X2',//,4X,'+ ',F10.3,
     +  'X3    + ',F10.3)
      ENDIF
*
      IF(NVAR.EQ.4) THEN
        WRITE(6,940)COEF(1),COEF(2),COEF(3),COEF(4),CIACT(1),CIACT(2),
     +  CIACT(3),CIACT(5),CIACT(6),CIACT(8),CFORD(1),CFORD(2),
     +  CFORD(3),CFORD(4),CONST
 940    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X4**2',//,4X,'+ ',F10.3,'X1*X2 + ',F10.3,
     +  'X1*X3 + ',F10.3,'X1*X4 +',F10.3,'X2*X3',//,4X,'+ ',F10.3,
     +  'X2*X4 + ',F10.3,'X3*X4 + ',F10.3,'X1    +',F10.3,'X2 ',//,4X,
     +  '+ ',F10.3,'X3    + ',F10.3,'X4    + ',F10.3)
      ENDIF
*
      IF(NVAR.EQ.5) THEN
        WRITE(6,950)COEF(1),COEF(2),COEF(3),COEF(4),COEF(5),CIACT(1),
     +  CIACT(2),CIACT(3),CIACT(4),CIACT(5),CIACT(6),CIACT(7),
     +  CIACT(8),CIACT(9),CIACT(10),CFORD(1),CFORD(2),CFORD(3),
     +  CFORD(4),CFORD(5),CONST
 950    FORMAT(3X,'Y= ',F10.3,'X1**2 + ',F10.3,'X2**2 + ',F10.3,
     +  'X3**2 +',F10.3,'X4**2',//,4X,'+ ',F10.3,'X5**2 + ',F10.3,
     +  'X1*X2 + ',F10.3,'X1*X3 +',F10.3,'X1*X4',//,4X,'+ ',F10.3,
     +  'X1*X5 + ',F10.3,'X2*X3 + ',F10.3,'X2*X4+'
     +  ,F10.3,'X2*X5',//,4X,
     +  '+ ',F10.3,'X3*X4 + ',F10.3,'X3*X5 + ',F10.3,'X4*X5 +',F10.3,
     +  'X1',//,4X,'+ ',F10.3,'X2    + ',F10.3,'X3    + ',F10.3,
     +  'X4    +',F10.3,'X5 ',//,4X,'+ ',F10.3)
      ENDIF
*
      WRITE(6,*)
      WRITE(6,*)' --VALUE OF INDEPENDENT VARIABLES AT OPTIMAL POINT--'
      WRITE(6,*)
      DO 960 I=1,NVAR
        WRITE(6,970) I,X(I)
 960  CONTINUE
 970  FORMAT(11X,'X',I2,'=',F16.6,/)
*
      WRITE(6,980) Y
 980  FORMAT(3X,'--VALUE OF RESPONSE VARIABLE AT OPTIMAL POINT--',//,
     +13X,'Y=',F16.6,//)
      WRITE(6,*)
      WRITE(6,*)
 990  CONTINUE
      CALL SKIP(24)
 1000 WRITE(*,*) '                 DO YOU WANT DATA TO PLOT CONTOURS?'
      CALL SKIP(2)
      WRITE(*,*) '                               "1" = YES'
      WRITE(*,*) '                               "0" = NO'
      CALL SKIP(5)
      READ(*,*,ERR=1000) IANS
      IF (IANS.NE.0.AND.IANS.NE.1) GO TO 1000
      IF (IANS.EQ.0) GO TO 1290
*
*                  PLOT CONTOURS
*
 1010 CALL SKIP(24)
      WRITE(*,*)'                        CONTOUR ANALYSIS'
      CALL SKIP(5)
      DO 1020 I=1,5
        Z(I)=0.0
        VMAX(I)=0.0
        VMIN(I)=0.0
        CONTOUR=0.0
        DELTA(I)=0.0
 1020 CONTINUE
      WRITE(*,*)'            WHAT VALUE CONTOUR DO YOU WANT TO PLOT?'
      CALL SKIP(7)
      READ(*,*) CONTOUR
      CALL SKIP(24)
      DO 1090 I=1,NVAR
 1030   WRITE(*,1040) I
 1040   FORMAT(10X,'WHAT IS THE MINIMUM VALUE OF X',I1)
        CALL SKIP(2)
        READ(*,*,ERR=1030) VMIN(I)
 1050   WRITE(*,1060) I
 1060   FORMAT(10X,'WHAT IS THE MAXIMUM VALUE OF X',I1)
        CALL SKIP(2)
        READ(*,*,ERR=1050) VMAX(I)

 1070   WRITE(*,1080) I
 1080   FORMAT(10X,'WHAT IS THE DELTA VALUE OF X',I1)
        CALL SKIP(2)
        READ(*,*,ERR=1070) DELTA(I)
        CALL SKIP(24)
 1090 CONTINUE
 1095 WRITE(*,*)'                   DO YOU WANT TO PRINT RESULTS?'
      CALL SKIP(2)
      WRITE(*,*)'             "1"= SEND RESULTS TO PRINTER AND SCREEN'
      WRITE(*,*)'             "0"= SCREEN ONLY'
      CALL SKIP(5)
      READ(*,*,ERR=1095) IPRINT
      IF(IPRINT.NE.0.AND.IPRINT.NE.1) GOTO 1095
      CALL SKIP(24)
      IF(IPRINT.EQ.1) THEN
        WRITE(*,*)'                         TURN ON PRINTER'
        CALL SKIP(8)
        CALL CONT
        CALL SKIP(13)
        WRITE(*,*)'            SENDING PLOTTING DATA TO PRINTER'
        CALL SKIP(12)
      ENDIF
        Z(1)=VMIN(1)
        Z(2)=VMIN(2)
        Z(3)=VMIN(3)
        Z(4)=VMIN(4)
        Z(5)=VMIN(5)
*              SEND RESULTS TO SCREEN
      WRITE(*,1100)CONTOUR
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,1110) (VMIN(I),I=1,NVAR)
      WRITE(*,1120) (VMAX(I),I=1,NVAR)
      WRITE(*,1130) (DELTA(I),I=1,NVAR)
      WRITE(*,*)
      WRITE(*,1140)
```

```
*                 SEND RESULTS TO PRINTER
      IF(IPRINT.EQ.1) THEN
        WRITE(6,1100)CONTOUR
        WRITE(6,*)
        WRITE(6,*)
        WRITE(6,1110) (VMIN(I),I=1,NVAR)
        WRITE(6,1120) (VMAX(I),I=1,NVAR)
        WRITE(6,1130) (DELTA(I),I=1,NVAR)
        WRITE(6,*)
        WRITE(6,1140)
      ENDIF
 1100 FORMAT(10X,' VALUES [+/- .01] TO PLOT CONTOUR = ',F9.2)
 1110 FORMAT(2X,'MIN VALUE - ',5(F13.3))
 1120 FORMAT(2X,'MAX VALUE - ',5(F13.3))
 1130 FORMAT(2X,'DELTA VALUE-',5(F13.3))
 1140 FORMAT(11X,'Y',6X,'--INDEPENDENT VARIABLES-- ')
 1150 Y=0.0D0

*
      DO 1160 K=1,NVAR
        Y=COEF(K)*Z(K)**2 + CFORD(K)*Z(K) + Y
 1160 CONTINUE
      Y=Y+Z(1)*Z(2)*CIACT(1) + Z(1)*Z(3)*CIACT(2) + Z(1)*Z(4)*CIACT(3)
     +  + Z(1)*Z(5)*CIACT(4) + Z(2)*Z(3)*CIACT(5) + Z(2)*Z(4)*CIACT(6)
     +  + Z(2)*Z(5)*CIACT(7) + Z(3)*Z(4)*CIACT(8) + Z(3)*Z(5)*CIACT(9)
     +  + Z(4)*Z(5)*CIACT(10) + CONST
      CHECK= Y-CONTOUR
      IF (CHECK.LE.0.01.AND.CHECK.GE.-0.01) THEN
        WRITE(*,1170) Y,(Z(I),I=1,NVAR)
        IF(IPRINT.EQ.1) WRITE(6,1170) Y,(Z(I),I=1,NVAR)
 1170   FORMAT(1X,F13.2,5F13.3)
      ENDIF
      IF (NVAR.LT.5) GO TO 1180
      IF(Z(5).GE.VMAX(5)) GO TO 1190
      Z(5)=Z(5)+DELTA(5)
      GO TO 1150
 1180 IF (NVAR.LT.4) GO TO 1200
 1190 Z(5)=VMIN(5)
      IF(Z(4).GE.VMAX(4)) GO TO 1210
      Z(4)=Z(4) + DELTA(4)
      GO TO 1150
 1200 IF (NVAR.LT.3) GO TO 1220
 1210 Z(4)=VMIN(4)
      IF (Z(3).GE.VMAX(3)) GO TO 1230
      Z(3)=Z(3) +DELTA(3)
      GO TO 1150
 1220 IF (NVAR.LT.2) GO TO 1240
 1230 Z(3)=VMIN(3)
      IF(Z(2).GE.VMAX(2)) GO TO 1250
      Z(2)=Z(2) + DELTA(2)
      GO TO 1150
 1240 IF (NVAR.LT.1) GOTO 1260
 1250 Z(2)= VMIN(2)
      IF (Z(1).GE.VMAX(1)) GO TO 1270
      Z(1)=Z(1) + DELTA(1)
      GO TO 1150
 1260 WRITE(*,*) '  # VARIABLES LESS THAN 1--ERROR'
      STOP
 1270 CALL CONT
 1280 CALL SKIP(24)
      IF (IPRINT.EQ.1) WRITE(6,*)
      IF (IPRINT.EQ.1) WRITE(6,*)
      WRITE(*,*)'                DO YOU WANT TO PLOT ANOTHER CONTOUR?'
      CALL SKIP(2)
      WRITE(*,*) '                            "1" = YES'
      WRITE(*,*) '                            "0" = NO'
      CALL SKIP(5)
      READ(*,*,ERR=1280) IANS
      IF (IANS.NE.0.AND.IANS.NE.1) GO TO 1280
      IF (IANS.EQ.1) GO TO 1010
      CALL SKIP(24)
*
 1290 CALL SKIP(24)
 1300 WRITE(*,*)'             DO YOU WANT TO ANALYZE ANOTHER SURFACE?'
      CALL SKIP(2)
      WRITE(*,*) '                          "1" = YES'
      WRITE(*,*) '                          "0" = NO'
      CALL SKIP(5)
      READ(*,*,ERR=1300) IANS
      IF (IANS.NE.0.AND.IANS.NE.1) GO TO 1300
      IF (IANS.EQ.1) GO TO 50
      CALL SKIP(24)
      STOP
*
      CALL SKIP(24)
      END
*
*                  SUBROUTINE CONT- USED TO HALT OPERATIONS UNTIL USER
*                  IS READY TO CONTINUE
*
      SUBROUTINE CONT
      CHARACTER*1 ANS,BLK
      DATA BLK/' '/
      ANS=BLK
      WRITE(*,10)
 10   FORMAT(/,'                       To continue, press RETURN key')
      READ(*,20) ANS
 20   FORMAT(A1)
      RETURN
      END
*
*                  SUBROUTINE SKIP- SKIPS "N" LINES ON SCREEN DISPLAY
*
      SUBROUTINE SKIP(N)
      DO 10 I=1,N
         WRITE(*,*)' '
   10 CONTINUE
      RETURN
      END
*
```